Mastering™
Red Hat®
Enterprise Linux® 3

Michael Jang

SYBEX ®

Mastering™
Red Hat® Enterprise Linux® 3

Michael Jang

SYBEX®

San Francisco London

Associate Publisher: Joel Fugazzotto

Acquisitions Editor: Elizabeth Peterson

Developmental Editors: Brianne Hope Agatep, Maureen Adams

Production Editor: Erica Yee

Technical Editor: Elizabeth Zinkann

Copyeditor: Kim Wimpsett

Compositor: Maureen Forys, Happenstance Type-O-Rama

Proofreaders: Laurie O'Connell, Nancy Riddiough

Indexer: Ted Laux

Book Designer: Maureen Forys, Happenstance Type-o-Rama

Cover Designer: Design Site

Cover Illustration: Jack T. Myers, Design Site

Library of Congress Card Number: 2004108201

ISBN: 0-7821-4347-4

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

To the young widows and widowers everywhere: our lives will never be the same. But life can be good again. For online help and resources for younger widows and widowers, see `www.youngwidow.org`, `www.fortnet.org/Widownet`, `www.groww.org`, and `www.ywow.org`.

My dear Nancy, I miss you. I feel joy as your spirit lives on through me.

Acknowledgments

It almost takes a village to create a computer book. Elizabeth Peterson had the vision to propose this book, the first major work dedicated to the Red Hat distribution customized for the enterprise. Brianne Agatep and Maureen Adams guided the development of this book from start to finish, marvelously making sure it stayed on track. Erica Yee did a wonderful job keeping the book moving forward.

This book could not go to press without the dedication and hard work of the other members of the team, including Maureen Forys and Ted Laux.

Most importantly, to this book, and to finding new life, I give special thanks to Elizabeth Zinkann, technical editor extraordinaire, logical Linux catalyst, and great friend. Not only is she the most Linux-savvy technical editor that I've ever encountered, she has been there to listen and help as I've worked through my grief. Here's to the coming (we hope) World Series between the Cubs and the Red Sox!

It does take a community to raise an operating system. I thank the thousands of developers around the world who donate their time to building Linux into an operating system that is challenging a monopoly.

On a personal note, thank you, Donna. Thank you for being there for me. Thank you for helping me to understand that we will always miss our dearly departed mates. Thank you for inspiring me to find life and love again, and hopefully to a new home of our own soon. You are the love of my new life.

I hope; therefore I can live.

Contents at a Glance

Contents

Introduction

According to *Forbes* (May 20, 2004), "Linux represents the biggest threat (that) Microsoft has ever faced. No wonder IBM is spending billions to promote it." Naturally, IBM is promoting Linux in the enterprise, and that may, in the title of this article, "Kill Bill."

Red Hat Enterprise Linux is the leading Linux distribution in the Enterprise. In this book, we give you the help you need to use Red Hat Enterprise Linux 3 productively in business and in life—in the enterprise or simply as a workstation on your desktop.

Linux is inexpensive. Linux is reliable. Linux is secure. With Linux, you can get the computing applications that you need—for a fraction of the cost of other operating systems. You need not worry about licensing fees. You can build a custom solution with the tools at hand.

In this time of stagnant budgets in information technology, the corporate world is getting more bang for the buck by moving toward Linux. Leading names in the financial sector, such as Goldman Sachs, Merrill Lynch, and Morgan Stanley, are moving toward Linux. Big online companies such as Amazon and Google use Linux to power their systems. IBM and Hewlett-Packard are generating billions of dollars of revenue from Linux. The list goes on.

While the heart of Linux is the command-line interface, Red Hat has developed a series of excellent graphical tools to help the administrators of other operating systems make the transition. Linux is built for networking. It is customized for TCP/IP, the language of the Internet.

Red Hat Enterprise Linux is the most popular large-scale Linux distribution. It includes applications such as office suites and specialized services that can easily cost hundreds of dollars per computer.

Linux is about freedom of choice. You can download "rebuilds" of Red Hat Enterprise Linux 3 for free. You can purchase "rebuild" CDs for a nominal fee from third parties. You can purchase it, with support and documentation from Red Hat. We explain each of these options at the end of this introduction. But no matter which version you are working with, this book will help you get the most from Red Hat Enterprise Linux.

What's in This Book

I've divided this book into eight parts, each addressing a different set of skills that can help you and your enterprise become productive in Red Hat Enterprise Linux. You can read this book from cover to cover, or use it as a resource when you need to know more about a specific skill.

Installing Red Hat Enterprise Linux In Chapter 1 we explain the roles that Linux can play as a desktop, as a small business server, and as a server for the enterprise. If you're planning to

install Linux on multiple computers, you'll want to read Chapter 2 carefully, because you need to be sure your hardware is ready for Linux. While Chapter 3 focuses on installing Red Hat Enterprise Linux locally using the graphical user interface, Chapter 4 shows you how you can install Linux over a network. In Chapter 5, we show you how to automate the installation process, which can be a great help if you're going to install Red Hat Enterprise Linux on a group of computers.

Linux Fundamentals To learn Linux in-depth, you need to know how to use the command-line interface. Once you learn how to navigate the file system in Chapter 6, the command-line interface can be your friend. In Chapter 7, we guide you through the skills you need to organize Linux filesystems. Once you've read Chapter 8, you'll know how to make the command-line shell work for you.

Basic Linux Administration Because Linux is built for networking, it is also built with a number of administrative tools. Administrators of this multiuser system need to know how to create, organize, and manage users and groups (Chapter 9). We show you how to use the Red Hat Package Manager and the Red Hat Update Agent (`up2date`) to install, upgrade, and manage applications securely (Chapter 10).

As an administrator, you'll need to go "under the hood" with the boot process (Chapter 11) and the Linux kernel (Chapter 12). You'll also want to know how to automate, manage, and troubleshoot basic services (Chapter 13), as well as back up your system (Chapter 14).

Basic Linux Networking Linux is built on TCP/IP, the language of the Internet. We guide you through the basics of TCP/IP as it applies to Linux. You can learn about basic TCP/IP protocols in Chapter 15 and the commands you need to apply them to your local area network (LAN) in Chapter 16. And we guide you through the fundamentals of network security in Chapter 17.

Linux Network Services Linux is built to serve all of the computers on a network. As an administrator, you need to know how to configure remote access (Chapter 18). TCP/IP networks require domain names and IP addresses, which are organized in DNS and DHCP servers (Chapter 19). Users on a network will want to print (Chapter 20) and use e-mail (Chapter 21).

Linux File Sharing Services Users share files between their computers. There are a number of ways to share files in Red Hat Linux. You can set up an FTP server just for files. If you're administering a network of computers that are running Linux and other Unix-style operating systems, you can share directories with NFS servers (both FTP and NFS servers are covered in Chapter 22). If you're setting up a network, it helps to set up a single database of users and passwords. You can do this with either NIS or LDAP (Chapter 23). If your network includes Microsoft Windows computers, you can make your Linux computer look like a client or a server on that network (Chapter 24). Apache is the most popular web server on the Internet and is optimized for Linux (Chapter 25). Finally, many enterprise users work with databases such as MySQL (Chapter 26).

A Certification Primer Many readers learn Linux to improve their job prospects. Today, that goes hand in hand with Linux certification. The three major distribution-neutral Linux certification programs are CompTIA's Linux+ exams, SAIR's Linux Certified Professional and Administrator exams, and LPI's Level I exams. Chapter 27 provides an overview of these exams targeted at Linux users with six months to two years of experience. Chapter 28 focuses on the requirements for the Red Hat certifications: the Red Hat Certified Technician and the Red Hat Certified Engineer.

X Window Management Desktop users need the graphical user interface (GUI). While ordinary users should never have to tinker with the basic X Window configuration (Chapter 29), administrators must know how to make it sing. This is the foundation for the two major Linux GUI desktop environments: GNOME and KDE; you can install a number of useful applications with each environment, including multiple office suites on either desktop environment (Chapter 30).

Appendices This book may be just one part of your journey into the world of Linux. Appendix A includes a very brief list of available online resources. Appendix B includes a copy of the GNU General Public License, which governs the use of Linux.

Conventions Used in This Book

If you're new to the world of Sybex books, you need to know about a number of conventions that we use.

- Linux commands such as `ls` and files such as `/etc/passwd` within the main body of a paragraph are offset as `inline code`.
- Longer lists of commands and code are organized in separate lines. The command prompt is shown as a hash mark (`#`).

  ```
  # mkbootdisk 2.4.21-158
  ```

- Hash marks are also commonly used in a program file to indicate a comment; I've done my best to make the context clear.

  ```
  # System initialization
  ```

- Sometimes the code you enter depends on a variable such as the version number, in which case the code is italicized.

  ```
  # mkbootdisk kernel_version
  ```

- *Italics* generally represent new terms.
- If an item is in bold in code, it represents what you might type in at the command-line interface to get the given output:

  ```
  # /usr/lib/yp/ypinit -m

  At this point, we have to construct a list of the hosts which will run NIS
  servers. Enterprise3  is in the list of NIS server hosts. Please continue to add
  the names for the other hosts, one per line. When you are done with the list,
  type a <control D>.
  ```

- + signs indicate key combinations. For example, Ctrl+Alt+F2 means you should press these keys simultaneously.

- With URLs, I've omitted the `http://` and the trailing slash for brevity (and to prevent bad line breaks). For example, the home page of the Linux Documentation Project appears as `www.tldp.org`, where it technically should be `http://www.tldp.org/`. Fortunately, with the defaults in web browsers and server software such as Apache, this generally makes no difference.
- When we discuss the Linux GUI, the menu arrow ➢ points you to a choice from a menu or submenu. For example, Main Menu ➢ Graphics ➢ The Gimp tells you to click on the Main Menu button, navigate to the Graphics menu, and then select The Gimp.

NOTE *Notes, in general, provide additional information outside the flow of a topic.*

TIP *Tips, on the other hand, are intended to help you in everyday use, such as configuring an application.*

WARNING *Warnings may highlight dangers to an application, the operating system, your hardware, and more.*

Getting Red Hat Enterprise Linux 3

An official copy of Red Hat Enterprise Linux 3 can be a little expensive; however, the price is not for the software itself but for support. The least expensive version for the server is Red Hat Enterprise Linux 3 ES Basic Edition for the Intel x86 CPU, with a list price of $349. Fortunately, there are less expensive, even free, options available.

Almost all of what's included with Red Hat Enterprise Linux 3 is released under the GNU General Public License, as shown in Appendix B. Red Hat has released the source code for these packages and makes them available in RPM format.

Several groups have taken Red Hat's source code and developed their own "rebuilds" of Red Hat Enterprise Linux 3. They are built from Red Hat's own source code. They have been modified to remove Red Hat trademarks such as the Red Hat fedora.

If you cannot afford Red Hat Enterprise Linux 3, we recommend that you get a rebuild of this distribution. While Red Hat releases its distribution on four CDs, some of the "rebuilds" include the same software on three CDs. They are from the following sources:

- Community Linux (`www.caosity.org`) includes "rebuilds" of both Red Hat Enterprise Linux 2.1 and 3.
- White Box Enterprise Linux (`www.whiteboxlinux.org`) has created "rebuilds" as well.
- Tao Linux (`www.taolinux.org`) includes "rebuilds" developed by one of the Linux administrators at Alfred University.

They are available by download from their Web sites (and mirrors); some are available on CDs from commercial third-party sources such as Linux Central (`www.linuxcentral.com`).

Other groups may also have created "rebuilds" of Red Hat Enterprise Linux 3. They may also offer the CDs or even DVDs with the latest updates for a nominal fee. There is one more alternative; you can purchase one of the workstation versions of Red Hat Enterprise Linux: Red Hat Enterprise Linux 3 WS or Red Hat Professional Workstation.

They include all of the packages associated with Red Hat Enterprise Linux 3 ES except for a few servers, such as those associated with DNS and Apache. They also include limited levels of support from Red Hat (except for the server packages associated with ES or AS).

Downloading Red Hat Enterprise Linux 3

If you have a high-speed Internet connection such as a cable modem or DSL adapter, you can download the Red Hat Enterprise Linux installation CDs. If you've purchased an official copy, you can download the CDs over the Red Hat Network. With your purchase, you should have an account and instructions on how to download the CDs in ISO format. With a CD writer and appropriate media, you can then use the `cdrecord` command described in Chapter 13 to write the ISO to a CD.

Alternatively, if you want to download the CDs of one of the rebuilds, we recommend that you use an FTP client such as gFTP. Microsoft Windows users may use clients such as WS FTP or Cute FTP. The steps in any GUI FTP client should be similar.

NOTE *I tried downloading Red Hat Linux over a telephone modem once—it took nearly two full days to download the first installation CD. Once downloaded, the data was corrupt. If you connect to the Internet through a telephone modem, I strongly suggest that you get Red Hat Enterprise Linux from Red Hat or a "rebuild" from a third party.*

To download rebuilds of the Red Hat Enterprise Linux CDs, you'll need an FTP client, sufficient room on your hard drive (at least 2.1GB of free space for the installation CDs), and the information described below:

FTP site There are FTP sites and mirrors associated with each of the "rebuilds." Details are available on their websites. You may get a faster response from a mirror, especially if you're downloading from outside the United States of America. Just be aware that often a delay occurs between the release of a Red Hat Enterprise Linux version or update and its availability on a mirror FTP site.

Username and password Normally, FTP sites for downloading "rebuild" Red Hat Enterprise Linux 3 CDs allow anonymous access. On such sites, the username is **`anonymous`**, and the password should be your e-mail address (though it isn't required).

Directory on the FTP server The actual directory on the FTP server varies with the site that you're using. More information may be available on the "rebuild" Websites. Some browsing may be required.

Besides the installation CDs, you may also see other CDs of interest, which include Red Hat Linux documentation.

A CD writer The Red Hat Enterprise Linux 3 CDs (and "rebuilds") are on huge files. You'll need CD-writing software and a CD drive that is capable of writing data to writablecompact discs. Linux includes a number of good CD writing applications described in Chapter 14.

Obtaining Third-Party "Rebuild" CDs with the Red Hat Enterprise Linux 3 Installation Files

Not everyone has a high-speed Internet connection. In that case, it may be more practical to purchase the downloaded CDs from a third-party reseller. The cost of the rebuild CDs is typically around $10 USD.

A directory of these resellers is available online at `directory.google.com`; click Computers ➢ Software ➢ Operating Systems ➢ Linux ➢ Companies ➢ Resellers for a list.

Getting the Red Hat Enterprise Linux 3 Boxed Set

You can purchase a full version of Red Hat Enterprise Linux from `www.redhat.com` and many major computer retailers. The boxed set, when purchased from Red Hat, is considerably more expensive than the download version. For more information, navigate to `www.redhat.com/software/rhel/purchase/`.

There are other versions available with support, which we briefly describe in Chapter 1. For a full list, see `www.redhat.com/software`.

Tell Us What You Think

We wrote this book to meet your needs, and only you can tell us if we've succeeded. If there are topics you expected to find here that we haven't covered, or if you find any errors, let us know by going to the page for this book at `www.sybex.com` and choosing the Submit a Review link. Of course, if this book has helped you to work better and faster with Red Hat Enterprise Linux, or if there are features we've included that you particularly like, we'd like to hear about that too. Good or bad, we'll use your feedback to build an even better book next time.

Part 1

Installing Red Hat Enterprise Linux

In this Part, you will learn:

- **Chapter 1: Introducting Red Hat Enterprise Linux**
- **Chapter 2: Preparing Your Hardware**
- **Chapter 3: Installing Linux on a Stand-Alone System**
- **Chapter 4: Installing Linux over a Network**
- **Chapter 5: Kickstarting Linux**

Chapter 1

Introducing Red Hat Enterprise Linux

Linux is a better way to run your computers. It's reliable, secure, and flexible. It's surprisingly easy to install. It's easier to use than most people think. It's highly customizable. It's built for networking. Even with the price of Red Hat Enterprise Linux, it's cost effective. Most important for the enterprise, it provides control; unlike the alternatives, enterprises can customize Linux to do exactly what's required.

For many people, Red Hat is Linux. That isn't quite right. Linux is based on software developed by a worldwide community of volunteers. Much of the initial work was spearheaded by the Free Software Foundation (`www.fsf.org`). Originally it was developed as a clone of the Unix operating system. Today, it is so much more. It's evolving to meet the needs of a wide variety of people, such as aerospace engineers, movie makers, theoretical physicists, and consumers. Even Wal-Mart is selling computers with Linux.

Strictly speaking, Linux is just the kernel, the part of the operating system that allows your software and hardware to communicate. But, oh, what a kernel! You can customize it in thousands of ways and update it for new features. Properly configured, it can optimize the effective speeds on your computer.

Red Hat Enterprise Linux is the basic Linux operating system, optimized for business. It incorporates security features developed by the U.S. National Security Agency for the kernel. It also includes a number of applications, such as a fully featured office suite, graphics programs, and multimedia applications that can satisfy most users.

Linux is fast becoming the major alternative to Microsoft Windows. As a server, it includes all the tools you may need to configure and administer a wide variety of networks. It has the backing of some major companies, including, as of this writing, Oracle, Dell, and Hewlett-Packard. IBM has invested more than a billion dollars in Linux just in 2001. Hewlett-Packard received $2.5 billion of Linux-related revenue in 2003. More and more companies are adopting Linux—as a server and as a desktop operating system.

Note *For those who are dedicated to the Apple Macintosh, remember that the latest Mac OS X was developed from an operating system closely related to Linux, the Berkeley Standard Distribution (BSD).*

While no one company is behind Linux, you can still get world-class support. Red Hat offers support and updates for its Enterprise operating systems; other companies do as well. If you participate in the give and take of the Linux community, thousands of developers will bend over backward to help you. This chapter covers the following topics:

- Introducing Red Hat Enterprise Linux 3
- Basic hardware requirements
- A short history of Unix and Linux
- Exploring the kernel
- Why choose Linux?
- The role of a Linux computer

Introducing Red Hat Enterprise Linux 3

Red Hat Enterprise Linux 3 is more than just an operating system: It's a complete distribution. It includes a wide variety of commands, utilities, and applications. Installing additional software in packages from the CDs is easy. With the right downloads from the Internet, you can always keep your version of Red Hat Enterprise Linux up-to-date.

Basic Hardware Requirements

Table 1.1 shows the minimum hardware requirements associated with Red Hat Enterprise Linux 3. These requirements are not absolute; for example, I've run Red Hat Enterprise Linux 3 at the command-line interface with as little as 96MB of RAM. Chapter 2 describes other hardware requirements.

These minimums assume a stand-alone Linux computer with just a few services. If you want to install additional software, configure a graphical user interface (GUI), or set up a server, the requirements go up accordingly.

Table 1.1: Basic Hardware Requirements

Type	Minimum
CPU	Pentium-class Intel-compatible 32-bit
	Intel Itanium or AMD64 (Workstation or Advanced Server)
	IBM zSeries, iSeries, pSeries, S/390 (Advanced Server only)
RAM	Minimum supported RAM for Intel 32-bit compatible architecture: 256MB
Hard disk	554MB (not including swap space or other files); more for other types of installations, as described in Chapter 3

Exploring Red Hat Products

Several versions of Red Hat Enterprise Linux are available as of this writing. Each version includes additional features, such as CDs and support, for a price. The features I cite in this sidebar were available at the time of this writing. The latest prices and support features are available at `www.redhat.com/software/rhel/purchase/index.html`. Alternatively, you can also get freely available versions of Red Hat *Enterprise* Linux from third parties without support.

Red Hat Enterprise Linux ES (Entry-Level Server)

Red Hat Enterprise Linux 3 ES supports basic servers, and is optimized for entry- and department-level server applications. It's the ideal solution for more basic file, print, web, and mail services. It is designed to run on computers with one or two Intel-compatible 32-bit CPUs; unfortunately, it does not support computers with other CPUs as of this writing. It's configured for computers with up to 8GB of RAM. The Basic Edition includes one year of access to the Red Hat Network, downloadable ISOs, and quarterly ISO updates. The Standard Edition adds physical installation CDs, printed documentation, web- and telephone-based support for one year.

Red Hat Enterprise Linux AS (Advanced Server)

Red Hat Enterprise Linux 3 AS is designed and optimized for larger organizations as well as the datacenter. It's certified for use with an extensive array of enterprise-level applications. You can install this operating system on computers with up to 16 CPUs. It supports basic servers and is optimized for entry- and department-level server applications. It's designed to run on computers with seven different architectures (prices vary by architecture and support level): Intel 32-bit, Intel Itanium, AMD64, IBM zSeries, IBM iSeries, IBM pSeries, and IBM S/390. It's configured for computers with up to 64GB of RAM. The Standard Edition includes one year of access to the Red Hat Network, downloadable ISOs, quarterly ISO updates, physical installation CDs, printed documentation, and web- and telephone-based support for one year. The Advanced Edition includes a premium level of web- and telephone-based support 24/7/365, with a one-hour response time.

Red Hat Enterprise Linux WS (Workstation)

Red Hat Enterprise Linux Workstation includes all but about 20 server RPMs included with Red Hat Enterprise Linux ES. It's designed to run on computers with one or two Intel-compatible 32-bit CPUs; a version is also available for 64-bit Itanium and AMD CPUs. In either case, it's configured for computers with up to 4GB of RAM. The Basic Edition includes one year of access to the Red Hat Network, downloadable ISOs, and quarterly ISO updates. Also, associated web- and telephone-based support is available for 30 days. The Standard Edition adds physical installation CDs, printed documentation, and web- and telephone-based support for one year.

Red Hat Professional Workstation

Red Hat Professional Workstation includes all the software associated with Red Hat Enterprise Linux WS; however, it only supports (up to 2) Intel-compatible 32-bit CPUs. As of this writing, it includes 30 days of installation (not configuration) support, as well as Red Hat Network updates.

Continued on next page

Exploring Red Hat Products *(continued)*

Other Red Hat Products

Red Hat has other specialty operating systems. These include the high-security Stronghold Enterprise Secure Web Server, Cluster Suite, Content Management System, Developer Suite, and Portal Server.

Red Hat Linux 9

As described in the introduction, Red Hat Linux 9 Personal Edition includes three installation CDs, three source CDs, and a documentation CD. It includes the software you need to install Linux in the Personal Desktop, Workstation, Server, or Custom configurations. Red Hat Linux 9 Professional Edition includes the source code and supplementary applications on CD. Red Hat Enterprise Linux 3 was developed from Red Hat Linux 9. While it's the latest freely available Red Hat operating system, it is no longer supported by Red Hat. (You may be able to get support from the Fedora Legacy Project at `www.fedoralegacy.org`.)

Fedora Core

Red Hat no longer produces freely available versions of the Red Hat operating system. It now supports the Linux community through the Fedora Linux project. The first versions of this operating system have been released as Fedora Core 1 and 2. As you can tell from the web address, `fedora.redhat.com`, it's still closely associated with Red Hat. Future advances in Red Hat Enterprise Linux may be tested on Fedora Core.

Third-Party Rebuilds of Red Hat Enterprise Linux

As of this writing, several groups have built and organized the over 1,100 freely available *source* RPMs associated with Red Hat Enterprise Linux 3. It includes virtually all the same software associated with this distribution and is freely available for download. Naturally, it doesn't include support or updates from Red Hat. Those available at the time of this writing include the following:

- cAos—Community Linux: `www.caosity.org`
- White Box Enterprise Linux: `www.whiteboxlinux.org`
- Tao Linux: `www.taolinux.org`

New Features

Red Hat is constantly incorporating new features and updating software. Most important are updates to the latest kernel and services. The following list includes some of the major improvements incorporated into Red Hat Enterprise Linux 3:

- Greater scalability; support for up to 16 CPU and 64GB systems.
- Native Posix Thread Library, which improves performance on multithreaded applications.
- Linux kernel version 2.4.21; Red Hat has customized it with proven changes to the Linux 2.5 and Linux 2.6 kernels, as well as a number of updated drivers. These changes are sometimes known as *backports*.

- The Common Unix Print System (CUPS), now the default print server, replacing LPD. For more information, see Chapter 20.
- Apache 2.0.46, now the standard Red Hat Enterprise Linux web server. For more information, see Chapter 25.
- Samba 3.0, which supports the transparent use of Linux as a Primary Domain Controller (PDC) on a Windows NT network or as a member server on a Windows 2000/2003 Active Directory network.
- `iptables`, now the default firewall tool (described in Chapter 17).
- XFree86 version 4.3 includes support for additional graphics adapters. It also has experimental support for RandR, which is the X Resize, Rotate, and Reflect extension (`http://www.usenix.org/events/usenix01/freenix01/gettys.html`).

Red Hat has also configured several tools not found in other Linux distributions. You can start these tools from a command-line interface inside a GUI such as GNOME (GNU Network Object Model Environment) or KDE (K Desktop Environment), using a `redhat-config-*` command. For example, `redhat-config-samba` lets you configure Samba, the service that allows Linux to work on a Microsoft Windows network. Samba is discussed in detail in Chapter 24.

Basic Components

Linux can be broken down into a number of modules. The modular nature of Linux allows developers to work independently and more efficiently. They can reuse and reconfigure these modules to achieve different results. At least six categories of modules are associated with Linux: kernel, network, `init`, daemons, shells and utilities, and the X Window.

Kernel

The kernel is the most important part of any operating system. It allows Linux and any software you install to communicate with computer hardware. The kernel communicates with your hardware through dedicated device drivers. For example, when you mount a floppy drive, a specific kernel driver sends and receives messages to and from the floppy drive.

If you install new hardware and it isn't detected when you start Linux, you can add a driver module to your kernel, as described in Chapter 11. If you have to download a driver for your new hardware, you should also add that driver module to the kernel.

Other parts of the kernel manage the Linux filesystem as well as any data stored in such areas as your disk cache. The kernel is loaded into protected-mode memory when you start Linux. You can learn how to configure and compile the kernel in Chapter 12.

In response to customer demand, Red Hat has chosen to stay with the stable, proven Linux kernel version 2.4. As version 2.6 was just released at the end of 2003, we anticipate that Red Hat won't incorporate this latest kernel until it's proven, and is ready for Red Hat Enterprise Linux 4. However, we've described the features from kernel version 2.6 that Red Hat has backported into the Enterprise Linux kernel.

Network

Linux computers are most commonly organized in a client/server network. Some computers act as workstations, or clients, for users; others are servers, which control resources shared by multiple users on different workstations. In this type of network, clients ask servers for items they need, such as files or applications. In a Linux network, clients can even ask for X Window information. In other words, you can set up terminals on Linux clients that access their GUI data from a Linux server.

The network modules of the Linux operating system attempt to keep client/server communication running as smoothly as possible. Ideally, the connection between client and server is seamless. If your network is fast enough, your users won't be able to tell the difference between local and network services.

Because network modules are loaded in the same area as the kernel, their failure may mean that you have to reboot Linux. We cover the basics of Linux networking in Chapters 15–17.

init

In general, the only way to start a Linux program is with another Linux program. For example, you log into the Linux terminal program, known as `mingetty`. But something has to start the terminal program. When you boot Linux on your computer, the kernel loads and starts `init`. The `init` program then mounts your drives and starts your terminal programs. When you log in, the terminal program starts your command-line interface shell.

After Linux boots on your computer, `init` watches for anything that might shut down your computer, such as a power failure signal from an uninterruptible power supply (UPS) or a reboot command. Details of `init` and the governing `/etc/inittab` file are discussed in Chapter 11.

Daemons

Linux includes a series of services. These are programs that can run in the background and start as needed. Many Linux services are known as *daemons*. In Linux, several dozen daemons can run simultaneously, standing ready to start your network, serve web pages, print your files, or connect you to other Linux or Windows computers. Typical daemons include the following:

- Apache, the most popular web server on the Internet, also known as `httpd`. Apache is covered in Chapter 25.
- Samba (also known as `smbd`), the network service that allows Linux to talk to Microsoft Windows computers. Samba is covered in Chapter 24.
- A printer daemon that manages communication with your printers. The CUPS daemon is `cupsd`; it's covered in more detail in Chapter 20.

We discuss various Linux daemons in detail throughout this book.

Tip *Case matters in Linux. For example, the acronym for the Common Unix Print System is CUPS; the associated daemon is* `cupsd`.

Shells and Utilities

Any Linux program or utility that talks to the kernel is a user-mode program, which consists of shells and utilities. User-mode programs don't communicate directly with your hardware (that's a job for the kernel). In other words, these programs can crash without affecting the basic operation of the Linux operating system. The three basic types of user-mode programs are as follows:

- *Login* programs associate a user ID with a user's shell and other personalized settings, such as with the X Window and web browsers.
- *Shell* programs act as Linux command interpreters. The most common Linux shell is known as *bash*, short for the Bourne Again Shell.
- *Utilities* are small-scale commands used inside a shell.

The basics of the bash shell and associated commands are covered in Chapters 6–8.

X Window

Linux builds the GUI from different program modules. GUI window managers, such as GNOME and KDE, as well as all GUI applications, are built on the foundation of the X Window. The basics of the X Window and associated applications are covered in Chapters 29–30.

A Short History of Unix and Linux

Linux was developed as a clone of Unix. In other words, the developers of Linux built their system without using the programming instructions, also known as the *source code*, used to build Unix. Because Linux is a Unix clone, you can use most of the same command-line commands on either operating system.

Although it would've been easier to adapt Unix for the personal computer, important historical reasons lie behind the development of Linux. And the way Linux was developed drives the way Linux developers, companies, and users work today.

Unix and the Coming Internet

Computers were once quite expensive. They were the domain of universities and larger corporations. There was a lot of demand for these early computers; to support this demand, a number of computer scientists developed the concept of *time-sharing*, where multiple users are connected to the same computer simultaneously.

Even though computers have become more powerful and less expensive, we've returned to this notion of time-sharing. Today, administrators are quite familiar with the concept of the time-sharing system: It's now known as the *multiuser server*. One network often includes multiple servers; your username may be the same across all these servers. In fact, it's fair to say that we're all time-sharing users on the biggest network of all—the Internet.

The following sections chronicle some of the developments that occurred along the road to Linux.

Multics

One of the early time-sharing projects was Multics (Multiplexed Information and Computing Service), a joint project between MIT, AT&T's Bell Labs (now Lucent Technologies), and General Electric. Although Bell Labs withdrew from the project in 1969, two of their developers, Ken Thompson and Dennis Ritchie, still had an itch for what would become the multiuser operating systems we know today.

Unix

Thompson and Ritchie continued development work through the early 1970s. Perhaps the key to their success was their development of the C programming language for writing the kernel and a number of basic commands, including those in the Bourne Again Shell, also known as bash.

When Unix was developed in 1969, AT&T was a regulated monopoly in the United States. Various court and regulatory rulings and agreements kept AT&T out of the computer business.

In 1974, AT&T distributed Unix to the University of California for the cost of the manuals and tapes. It quickly became popular at a number of universities. Nevertheless, AT&T wasn't allowed to make a profit from it.

A Cooperative Environment

Bell Labs has a history of groundbreaking research. The company had some of the best minds in the world working on fundamental problems. Bell Labs wanted the goodwill of the academic community. Since AT&T wasn't allowed to make money from software, it kept the license for Unix and distributed the operating system with source code to universities for a nominal fee. In exchange, AT&T's lawyers insisted that the license explicitly state that Unix came with no warranty. This release technique became known as *open source*.

The timing was good. Various universities adapted the Unix source code to work with three different kinds of computers available at the time: mainframes, minicomputers, and microcomputers.

At about the same time, the U.S. Department of Defense's Advanced Research Project Agency (ARPA) wanted to set up a nationwide communications network that could survive a nuclear war. Most universities on this ARPA network used Unix. TCP/IP was built on Unix and eventually became the communication protocol for the ARPAnet. The ARPAnet eventually developed into the Internet you know today. Unix and derivative clones, such as Linux, are critical parts of the Internet.

The AT&T Consent Decree

AT&T retained the license to Unix through the 1980s. When the U.S. government settled the AT&T antitrust suit in 1982, one of the provisions allowed AT&T to go into the computer business. This became known as the AT&T *consent decree*. At that point, AT&T was able to sell the Unix operating system and source code with all the protections associated with a copyright.

The programmers who used Unix wanted to keep the advantages of an open-source operating system. Unix programmers wanted the ability to customize the software. As academics, they wanted to share the results. The Unix users of the time had the high level of knowledge that made open-source software worthwhile.

Ironically, AT&T was never very successful at selling Unix and eventually sold the rights to the operating system. The direct successor to AT&T's version of UNIX is now owned by the SCO

Group, who once released Caldera Linux under the GPL. While access is now closed, SCO has ironically renamed the Caldera Linux distribution: "OpenLinux." The US legal system has not yet decided on whether the SCO Group or Novell owns the rights to the Unix source code. Novell owns SuSE, which developed the main alternative high-end distribution to Red Hat Enterprise Linux 3.

NOTE *The SCO Group recently filed suit against IBM over Unix. This is controversial in part because it has attacked the General Public License (GPL), under which so much Linux software has been released. While I think SCO's claims are worthless, the case is still pending as of this writing. It isn't even scheduled for trial until early 2005. The details are complex; there are multiple countersuits, and potentially a large number of groups is involved. A good source for the latest information is* `www.groklaw.net`.

Unix Alternatives

At the time, with their limited budgets, universities didn't have the money to purchase the now proprietary Unix, and they didn't want to have their academic freedoms limited by copyrights. Generally, academics are most comfortable when they can share all of their data. To this end, Douglas Comer developed Xinu (Unix, spelled backward) in 1983 to illustrate operating system structures in a classroom setting. In 1986, Andrew Tannenbaum developed Minix as a Unix clone and free alternative. Like Linux, Minix doesn't use Unix's source code, and therefore it doesn't infringe on any of AT&T's Unix copyrights.

Even before the consent decree, Bill Joy of the University of California worked on Unix. He also started work on the Berkeley Standard Distribution (BSD), which, like Unix, was released under an open-source style license. A number of BSD utilities were incorporated into later versions of Unix. In 1982, Joy became a cofounder of Sun Microsystems.

Several other operating systems are closely related to Unix, as shown in Table 1.2.

One telling trend is that a number of these companies are moving toward using Linux on many of their servers. While this book is based on the Intel 32-bit Red Hat Enterprise Linux kernel, different kernels are available for the AMD64, Intel Itanium, PowerPC, IBM S/390, and other IBM platforms.

TABLE 1.2: UNIX-STYLE OPERATING SYSTEMS

OPERATING SYSTEM	DESCRIPTION
AIX	The Advanced Interactive eXecutive operating system, developed by IBM; used with high-end CPUs such as Power4 and RS64 IV (64-bit PowerPC chips).
BSD	The Berkeley Standard Distribution, an open-source alternative to Linux.
HP-UX	Developed by Hewlett-Packard; version 11i is developed for 64-bit RISC and Itanium CPUs.
IRIX	Developed by Silicon Graphics for 64-bit CPUs.
Linux	The free operating system clone of Unix.
Solaris	Developed by Sun Microsystems for its UltraSPARC CPUs.
Tru64	Formerly known as Digital Unix, optimized for 64-bit CPUs.
UnixWare	The successor to AT&T's version of Unix, now owned by the SCO Group.

The General Public License

Stallman developed the GPL to bring the advantages previously available with Unix to the general software community. He wanted to develop a license that would protect software from anyone who would hide its source code. GNU software is licensed under the GPL. While you can read the GPL in Appendix B, you can also learn three of the basic principles behind the GPL.

- All GPL software must be distributed with a complete copy of the source code. The source code must include clear documentation.
- Any software added to GPL software must also be clearly documented. If the new software interacts with the GPL software, the package as a whole must be distributed as GPL software.
- Any GPL software comes without a warranty.

The Free Software Foundation

Some of the work of the academic community on cloned Unix software eventually developed into a serious rebellion. In its early stages, it was led by Richard Stallman and his Free Software Foundation (FSF). (For more information, see their website at `www.fsf.org`.)

Stallman started work on the GNU's Not Unix (GNU) project in 1984. He summarized the focus of the FSF in his introductory Usenet message: "I consider that the golden rule requires that if I like a program I must share it with other people who like it." Stallman's purpose was to set up a group where the free sharing of software would be strongly encouraged. To realize his dream, Stallman needed an operating system, free of the code that was then copyrighted by AT&T.

The FSF developed the General Public License (GPL) to build a body of free software protected from those who would use it to create proprietary closed-source systems. This same license still protects Linux today; you can read it in Appendix B.

By 1991, the FSF had cloned all of the major components of a Unix-style operating system except the kernel.

Linus Develops a Kernel

In 1991, Linus Torvalds was a graduate student in Finland. He wasn't happy with the operating systems available for his new computer with a 386 CPU. So he put together a kernel to allow some operating system components to communicate with computer hardware. By 1995, several companies assembled Linus's kernel with the GNU software of the FSF to produce the first Linux distributions.

NOTE *Richard Stallman and the people of the FSF believe that the Linux operating system is more properly known as GNU/Linux because it combines a large number of GNU-licensed programs, commands, and utilities with one Linux kernel.*

Exploring the Kernel

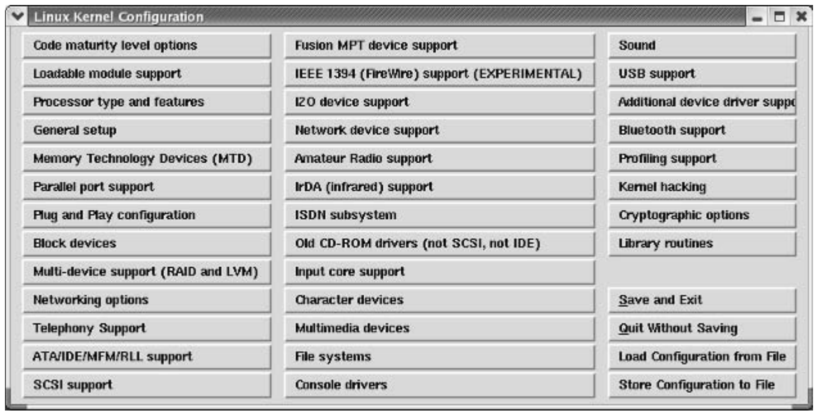
Life in any operating system begins and ends with the kernel. When properly configured, any operating system can work like a wonderful ballet where hardware is ready just when you need it. When problems crop up, the kernel can slow or stop your computer.

With the Linux kernel, you can configure hardware, filesystems, networking support, and more. Hardware drivers can be configured within the kernel or as separate modules.

Configuring the Kernel

If you ever need to reconfigure your kernel, you'll become familiar with the Linux Kernel Configuration menu shown in Figure 1.1. As you can see, there are a number of different hardware components, such as SCSI and USB devices, that you can configure through the kernel. Each of the buttons shown in the menu opens individual submenus.

Figure 1.1
Linux Kernel Configuration



You can also see some kernel options not directly associated with hardware, such as Networking Options and Code Maturity Level Options. For example, in the Networking Options menu, you can set up Linux to work with different network protocols. You'll find detailed information on reconfiguring the kernel in Chapter 12.

The */proc* Filesystem

The `/proc` directory is a virtual filesystem stored in your RAM. It documents the way the Linux kernel interacts with your computer. A number of these files document how the Linux kernel reads your hardware. When you read the right file, you can find hardware settings for different components. You can find more information on `/proc` in Chapter 11.

Modular or Monolithic

You can set up every hardware driver within the main part of the Linux kernel. This would be a *monolithic* kernel. But for most configurations, there are many hundreds of hardware drivers. If you put them together into one kernel file, the sheer size of the hardware drivers can overload your system.

It's usually more efficient to configure a *modular* kernel. Various kernel modules, normally associated with various hardware components, are loaded after Linux starts on your computer. Figure 1.2 shows an example from my Red Hat Enterprise Linux 3 server.

FIGURE 1.2
Linux modules

```
[root@Enterprise3 linux-2.4]# lsmod
Module                  Size  Used by    Not tainted
nfs                    92912   1  (autoclean)
smbfs                  44528   1  (autoclean)
ide-cd                 35680   0  (autoclean)
cdrom                  33696   0  (autoclean) [ide-cd]
nfsd                   85456   8  (autoclean)
lockd                  59856   1  (autoclean) [nfs nfsd]
sunrpc                 85692   1  (autoclean) [nfs nfsd lockd]
autofs                 13364   1  (autoclean)
pcnet32                18080   1
mii                     3976   0  [pcnet32]
crc32                   3712   0  [pcnet32]
ipt_REJECT              4632   1  (autoclean)
ipt_state               1080   2  (autoclean)
ip_conntrack           27304   1  (autoclean) [ipt_state]
iptable_filter          2412   1  (autoclean)
ip_tables              15776   3  [ipt_REJECT ipt_state iptable_filter]
floppy                 58160   0  (autoclean)
microcode               4724   0  (autoclean)
loop                   12120   0  (autoclean)
lvm-mod                64704   1
keybdev                 2976   0  (unused)
mousedev                5524   1
hid                    22212   0  (unused)
input                   5888   0  [keybdev mousedev hid]
usb-uhci               26412   0  (unused)
usbcore                79392   1  [hid usb-uhci]
ext3                   91592   3
jbd                    52336   3  [ext3]
raid1                  14988   2
[root@Enterprise3 linux-2.4]#
```

As you can see, there are hardware modules, such as `usbcore`, to support USB hardware. There are also software modules, such as `smbfs`, to support the Samba filesystem. For more information on managing kernel modules, see Chapter 11. If you want to make sure that your kernel is modular, see Chapter 12.

Why Choose Linux?

Linux is most frequently compared to Microsoft Windows. Linux is also replacing other Unix-style operating systems described earlier in Table 1.2. Four factors make Linux a better choice for many users and organizations: control, cost, reliability, and support.

Control

Linux provides a great deal of control of your computers. The Red Hat Package Management (RPM) system makes it easy to add the software you need. For more information on RPM packages and the `rpm` command, see Chapter 10. The `redhat-config-packages` tool, also described in Chapter 10, makes this process of software management even easier.

You can optimize the Linux kernel using the techniques discussed in Chapter 12. An optimized kernel makes everything faster in Linux, from the boot process to networking. With the right techniques, you should always have an easily accessible working kernel; in contrast, small errors when changing the Microsoft Windows Registry can be disastrous.

Linux is easily upgradeable. You can keep an older version of Linux up-to-date with the latest in kernels, applications, and other software. The `rpm` and `up2date` tools described in Chapter 10 help you with this process.

Cost

Despite the price, the software associated with Red Hat Enterprise Linux 3 doesn't have to cost you anything. The price charged by Red Hat defrays its cost of support and updates. You can download free "rebuilds" of this operating system from one of the third-party providers described earlier in the "Exploring Red Hat Products" sidebar.

The price Red Hat has set includes support. There is no licensing fee as this is still almost all GPL software. This cost difference can still be significant when compared to the thousands of dollars required just to license many other Unix-style operating systems—as well as Microsoft operating systems.

The price of the operating system isn't the only cost. Other costs include the time required for installation, configuration, and internal support.

Red Hat Enterprise Linux includes one additional cost advantage: The CDs are loaded with a number of fully featured applications. For example, OpenOffice.org is a fully featured office suite, with all of the features that most users could ever want. Red Hat includes several other free applications that can save you hundreds of dollars on each computer.

Installation

As you install Linux over the next few chapters, you'll learn that the process isn't difficult. If you're installing Red Hat Enterprise Linux on a group of computers, you can use the Kickstart techniques described in Chapter 5 to automate the installation process.

Since Red Hat Enterprise Linux 3 can be installed on most computers without a problem, the discussion of hardware in Chapter 2 may seem extreme. However, if you're an administrator responsible for installing Linux on several high-priced servers, mistakes can quickly get expensive.

Configuration

To make any operating system useful, you need to install and configure it. Whether you're configuring a server for your department, corporate network, or global website, the basic configuration process remains the same. Linux has always had the command-line tools with the flexibility to satisfy most Linux gurus.

With the `redhat-config-*` tools described throughout the book, Red Hat Enterprise Linux also offers the graphical tools that can help administrators of more graphical operating systems make the transition.

Reliability

Linux is reliable. It's common to see reports of Linux servers that run for several months at a time without reboots. Imagine never having to reboot your computer after installing new software. Imagine being able to stop a runaway program without rebooting your computer. That's the power of Linux.

Linux isn't perfect. Mistakes happen. I describe troubleshooting techniques throughout this book. If you ever have a problem booting Linux, you can rescue most systems with your Red Hat Enterprise Linux installation CD (without reinstalling Linux).

Support

There are a number of ways to get excellent support for your Linux system.. The support associated with Red Hat Enterprise Linux answers the concerns of many administrators and IT managers, who want a single source of corporate support. If you purchase Red Hat Enterprise Linux, you can get some of the support you need directly from Red Hat. You can purchase additional support from third-party vendors such as Linuxcare (`www.linuxcare.com`). Some of the large companies behind Linux, such as IBM and HP, also provide support for Red Hat Enterprise Linux as installed on their systems.

There are two bonus sources of support for Linux. Because Linux is open source, administrators can often fix many problems. If you're working with a closed-source system, you can't even "look under the hood."

Since Linux is developed by a community, there are many in that community who are anxious to make their name by solving new problems. Their insights are available online. It's quite possible that the answer to your problem is already available in the Internet newsgroup database, accessible through `groups.google.com`.

The Role of a Linux Computer

You can configure Red Hat Enterprise Linux 3 as a server or as a workstation. Linux is flexible; you can install it on many older computers that you may otherwise have to scrap.

Red Hat includes a number of additional programs and applications that enhance what Linux can do as a commodity server, in the enterprise, and even on the desktop.

Linux as a Server

Linux is built for networking. You can set it up as a server to manage many different kinds of resources for your network. Table 1.3 lists just a few of the Linux services you can configure. Many of these services have their own individual daemons. Others are associated with the Extended Internet Services Daemon (`xinetd`) described in Chapter 18.

TABLE 1.3: LINUX SERVER SERVICES

Service	Description	Chapter
`crond`	Runs scripts on a schedule	13
`cups`	Manages the Common Unix Print System (CUPS)	20
`httpd`	The Apache web server	25
`mysqld`	The MySQL server	26
`named`	The Domain Name System service	19
`nfs`	A Network File System server	22
`sendmail`	A common e-mail transport agent	21
`slapd`	A Lightweight Directory Access Protocol service	23

Table 1.3: Linux Server Services *(continued)*

Service	Description	Chapter
`smb`	Samba, which makes Linux computers members of Microsoft Windows networks	24
`squid`	A web caching proxy service	25
`sshd`	Secure Shell	18
`vsftpd`	The Very Secure FTP Daemon	22
`xinetd`	The Extended Internet Services Daemon	18
`ypserv`	A Network Information Service server	23

It's common to install Linux on older computers. You can set up a Linux computer as a server with limited functionality. In many cases, this doesn't require a great deal of RAM or hard disk space. For example, you could set up a Linux computer as a modern print server or a firewall. You wouldn't have to purchase dedicated hardware for these purposes.

Linux on the Desktop

Linux is a serious alternative on the desktop. As you'll see in Chapter 30, Linux provides essentially the same basic GUI applications and configuration tools that you can find in any version of Microsoft Windows.

In addition, three major office suites are available that you can use in place of Microsoft Office. Mozilla and Konqueror are fully featured web browsers; alternatively, you can still install Netscape or Opera on Linux. Evolution provides an alternative to Microsoft Outlook.

People are taking a serious look at Linux on the desktop. As of this writing, Wal-Mart is selling four different computers with Lindows (`www.lindows.com`), a version of Linux that's customized to run a number of Microsoft Windows applications. Linux is getting a serious look as a desktop alternative outside the United States.

Game manufacturers are creating ways to play on Linux. Tux Games is an online store (`www.tuxgames.com`) with a warehouse of interesting games. There's even a version of The Sims for Linux, courtesy of TransGaming Technologies (`www.transgaming.com`).

Applications available for Linux may not meet everyone's needs. In the personal finance area, GNUcash, in my opinion, does not compare well with the latest versions of Quicken. Other Linux personal finance programs are listed at `www.linuxlinks.com/Software/Financial/Personal_Finance/`.

If you need a few Microsoft Windows programs, multiple solutions are available. CodeWeavers' CrossOver Office (`www.codeweavers.com`) allows you to run Microsoft Office 97/2000, Quicken, Lotus Notes, and more. You can set up Microsoft Windows inside a virtual computer inside Linux, courtesy of VMware (`www.vmware.com`) or NeTraverse Win4Lin (`www.netraverse.com`).

Red Hat Enterprise Linux 3 Workstation

This book is focused on using Red Hat Enterprise Linux 3 as a server. However, Red Hat Enterprise Linux 3 Workstation provides a powerful desktop alternative that you can use with 32-bit and 64-bit Intel and AMD architectures. This type of power is suitable for higher-end "desktop" environments such as Computer-aided Design (CAD) stations and databases.

To this end, Red Hat has configured GNOME and KDE with a similar look and feel. These changes are documented in an "overlay" to these desktop environments, known as Bluecurve. I describe these desktop environments and associated applications briefly in Chapter 30.

Workstation users may be pleased with the wide array of applications that come with Red Hat Enterprise Linux 3. They include the following:

- OpenOffice.org, a fully featured office software suite.
- Mozilla and Konqueror, web browsers as fully featured as Microsoft Internet Explorer.
- Internet utilities such as Instant Messenger, news clients, remote desktops, and more. (In fact, I use the Linux Instant Messenger application to connect to the Microsoft Messenger network.)
- Multimedia applications that allow you to write CDs and even DVDs at full speed.

While the Red Hat desktop graphics utilities don't yet have the CMYK (cyan, magenta, yellow, and black) graphics software such as Paint Shop Pro, a number of movie studios do create animation and special effects on Linux computers.

NOTE *CMYK is a color model more popular in high-end graphics applications than the original RGB (red-green-blue) standard.*

Red Hat Enterprise Linux for Small Businesses

Red Hat Enterprise Linux can be a fantastic option for small businesses or organizations. You can install it on servers and workstations. It provides the scalability and control needed by a small growing business. Small businesses can purchase and install Red Hat Enterprise Linux 3 with support. Alternatively, they can download and use the freely available "rebuilds," or the work of the Fedora project. These alternatives use the same software developed by Red Hat, and they can keep the cost of the operating system to a bare minimum.

Red Hat Enterprise Linux computers are fairly easy to configure in a network, even if you need to connect them) to Microsoft Windows computers. You can even configure Red Hat Enterprise Linux as a PDC in a Microsoft Windows–style network. Once Samba is properly configured (see Chapter 24), other Microsoft computers won't be able to tell the difference.

With the right configuration, you can easily connect your network to the Internet. You can also protect your network from many of the ravages of the Internet with appropriate settings on your firewall and other network tools.

Red Hat Enterprise Linux for Bigger Business

Many corporations, governments, and educational institutions have already installed Red Hat Enterprise Linux to power their high-demand servers. With the certified options available for this distribution, you'll get a powerful enterprise tool for everything from Oracle databases to high-capacity secure web servers.

Amazon.com has saved millions by converting to Red Hat. Google runs its search engine databases on a cluster of more than 8,000 servers running Red Hat. This operating system is becoming more popular for other large organizations as well, such as BP, Kenwood, and MIT. In a Red Hat case study, Toyota actually found slightly *lower* support costs after converting its computers to Red Hat.

Summary

Linux was developed as a clone of Unix. The Free Software Foundation reverse-engineered most of the key components of Linux. Critical was Linus Torvalds's creation of the Linux kernel. Most of it is protected through the General Public License.

As Red Hat Enterprise Linux 3 is being released, businesses and governments want control over their operating systems. Because Linux is modular and highly configurable, it provides the support that organizations need to keep their costs to a minimum.

Reporting Problems

Linux is a work in progress. Developers are constantly adding and revising features for new software and hardware. It's possible that in your journey with Linux, you'll run into a problem or two. There are four ways to look for a solution:

Red Hat support If you've paid for an official copy of Red Hat Enterprise Linux, you can get the support as described at `www.redhat.com`. As of this writing, this includes anywhere from 30 days of basic installation and configuration support to a full year of more complete support.

Newsgroups As described earlier, many users bring up problems they have in different newsgroups. Google collected recent newsgroup messages into a searchable database at `groups.google.com`. If you want to post on a newsgroup, it's best to use a newsgroup reader such as those described in Chapter 30. Alternatively, you can post messages using Google's interface at `groups.google.com` (registration is required).

Mailing lists Red Hat has a series of mailing lists on different topics and versions; you can sign up at `www.redhat.com/mailing-lists/`. The developers of a number of different applications maintain their own mailing lists, which you can find on their websites.

Bugzilla If you're certain the problem is with Red Hat Enterprise Linux, you can submit a bug report to Red Hat. Navigate to `bugzilla.redhat.com/bugzilla`, and click Login or New Account. Create an account if you don't already have one. If you have an official copy of Red Hat Enterprise Linux, you should be able to use the account you created when you purchased support for this operating system. You can then search through the Bugzilla database to see if someone has already raised the issue with Red Hat. If not, and if you've exhausted the other resources, submit a bug report through the Red Hat Bugzilla system.

Red Hat Enterprise Linux 3 includes the same basic components as all other Linux distributions: the kernel, `init`, daemons, user-mode shells and utilities, network, and the X Window. It incorporates the latest changes to the Linux kernel, as well as improvements in printing, web services, and more. The `redhat-config-*` graphical tools make it easier for administrators of other operating systems to make the transition.

When looking at Linux, you should consider four factors: control, reliability, cost, and support. I believe that Linux has advantages in all four areas when compared to other operating systems.

Red Hat Enterprise Linux can play many roles in computing. Traditionally, it's used as a server, and functions well even on many older computers. Red Hat is adding tools that make it suitable as a high-powered workstation operating system. Such flexibility makes Red Hat a viable alternative for small businesses. Red Hat is also being used in the enterprise, on clusters of computers to meet the heaviest demands.

In the next chapter, we'll start looking at getting your computers ready for Red Hat Enterprise Linux. Installation often does proceed easily on most modern computers. However, if you're installing Red Hat Enterprise Linux on two or more computers on a network, mistakes can be painful. If you're responsible for installing Linux on a network, you need to know more about the hardware in your computers.

Chapter 2

Preparing Your Hardware

In most cases, installing Red Hat Enterprise Linux is a trouble-free process. If you're installing Red Hat Enterprise Linux on a new workstation or server, all you *probably* need to do is insert the installation CD in the correct drive, set your computer's BIOS to boot from the CD, restart your computer, and you're ready to go. The Red Hat Enterprise Linux installation program should start and detect most hardware automatically.

If you have a relatively new PC with at least an Intel-style Pentium-level CPU, and if you don't have the absolute latest in computer hardware, you may never have to worry about Linux drivers. While you should at least read the first sections on disk partitions, you may be able to skim much of this chapter.

However, suppose your workstation includes proprietary hardware without Linux drivers. Perhaps your server has hardware that's too new to have Linux drivers. Or you have a slightly older PC that's prone to hardware conflicts. Perhaps you're responsible for installing Linux on a network of computers where hardware problems can get expensive.

In that case, it pays to have a detailed list of hardware on your computers. Then you can review available lists of compatible hardware. With a little work, a perfect match isn't even required. With the right resources, even configuring the dreaded Winmodem is easier than you might expect.

Many users who are just learning Linux set their computers up in a "dual-boot" configuration, where they can start either Red Hat Enterprise Linux or Microsoft Windows (or even another operating system) during the boot process. Preparing a dual-boot on a computer that currently has only Microsoft Windows does take some work. This chapter covers the following topics:

- Creating hard disk partitions
- Configuring Microsoft and Linux with a 32-bit architecture
- Why worry about hardware?
- Finding compatible hardware
- Creating a hardware checklist
- BIOS tips
- Post-installation hardware configuration

Creating Hard Disk Partitions

If you're configuring a modern server, you're probably working with very large hard disks, especially when compared to a regular PC. But even the latest PC hard disks are now fairly large. In either case, partitions help you configure hard disks in manageable chunks. When configured correctly, partitions can help protect your system. For example, if someone overloads your FTP server with files, the right partitions ensure that your system still has room to run.

Alternatively, if you have a smaller hard disk (less than 4GB), you'll need to be efficient. If you overpartition a drive, you may not have enough space for certain types of additional files.

You can organize each physical hard disk into *primary*, *extended*, and *logical* partitions. The details depend on whether you're configuring a regular IDE (Integrated Drive Electronics) hard disk or a SCSI (Small Computer Systems Interface) hard disk.

Linux is organized into directories. You can mount different directories onto partitions according to the Filesystem Hierarchy Standard. We cover the FHS and typical partition configurations for Red Hat Enterprise Linux in Chapter 7.

Partition Styles

You can even configure different operating systems on the same hard disk, using different partitions. Each filesystem can be formatted to different filesystems, such as the default Third Extended Filesystem (ext3) or other popular server filesystems such as ReiserFS or XFS. In this vein, there are four ways to partition a hard drive:

Primary partition You can have up to four different primary partitions on a hard drive. One primary partition must be marked as "active" and typically includes a bootloader, such as the Grand Unified Bootloader (GRUB). If you mount a Linux directory on a primary partition, it is also known as a *volume*.

Extended partition If four partitions aren't enough, you can convert one of the primary partitions into an extended partition. You can then subdivide the extended partition into as many logical partitions as you need. But you can't mount a directory on an extended partition.

Logical partition You can subdivide an extended partition into logical partitions. While you can configure more during the installation process, Red Hat supports only 11 logical partitions on each physical drive. Although you can't set up a Linux directory in an extended partition, you can set up Linux directories on logical partitions configured within an extended partition. Therefore, logical partitions are also *volumes*. In the Microsoft world, these would be *logical drives*.

Swap partition In Linux, you'll want to set up a swap partition as an exclusive area for the virtual memory on your hard drive. Swap partitions aren't a different kind of partition per se; they can be mounted on a primary or logical partition. While the appropriate size of a swap partition is highly debatable, Red Hat generally recommends you set up a swap partition with twice the amount of memory in your RAM. Some suggest that at larger amounts of RAM (greater than 1GB), you may be able to manage with a swap partition that equals the amount of RAM on your system. Others suggest you need to configure server swap partitions with four times the amount of RAM on your system.

Since this depends on the demands of your particular network, pay attention to the `fdisk` utility described in Chapter 7. Learn how to create additional swap partitions as needed.

Partition Names

The Linux naming convention for hard disk partitions is straightforward. The naming system also applies to any CD that doesn't require a direct connection to a sound card. The first two letters of the name reflect the kind of disk you have. If you have a regular IDE hard disk, the letters are `hd`. If you have a SCSI hard disk, the letters are `sd`.

The third letter depends on your hard disk's position. The first hard disk is designated as `a`, the second disk is designated as `b`, and so on. In other words, if you have two different physical IDE hard disks attached to the primary controller, the second (slave) disk is known as `hdb`. In contrast, SCSI hard disk letters correspond to their designated ID numbers. For example, if you have two SCSI drives with IDs of 0 and 1, the SCSI drive with an ID of 0 is known as `sda`; the SCSI drive with an ID of 1 is known as `sdb`. For naming purposes, CD and DVD drives are also categorized as hard disks.

The character in the fourth position reflects how you've partitioned that disk. Because you can have up to four primary partitions, they are designated as 1, 2, 3, and 4. The first logical drive that you create is in position number 5, even if you have only one primary partition.

Every partition is associated with a Linux device file in the `/dev` directory. When you mount a directory on a partition, you're associating it with the device file. Some examples of different partition device files are shown in Table 2.1.

Table 2.1: Typical Partition Device Names

Name	Description
`/dev/hda3`	The third primary partition on the master hard disk on the primary IDE controller; depending on your configuration, it may also be an extended partition.
`/dev/sdc8`	The fourth logical partition on the third SCSI hard disk.
`/dev/hdb7`	The third logical partition on the slave hard disk on the primary IDE controller.
`/dev/sda1`	The first primary partition on the first SCSI hard disk.
`/dev/hdb`	Since there's no number, this refers to a CD or DVD drive attached as the slave on the primary IDE controller.
`/dev/sdc`	Since there's no number, this refers to a CD or DVD drive attached to the third position on a SCSI interface.

Configuring Microsoft and Linux with a 32-Bit Architecture

Generally, if you're installing Red Hat Enterprise Linux 3 on a server, you won't want to install it on the same computer with another operating system such as Microsoft Windows. If you need Microsoft Windows software on your network, it's best to have it available on a different physical computer. There are exceptions; someone who is converting a Primary Domain Controller (PDC) from Windows NT 4 to Linux may want a dual-boot of both systems.

If you're setting up Red Hat Enterprise Linux 3 as a workstation, you may want to install Microsoft Windows and Linux on the same computer. Good software is available that, as of this writing, works only on Microsoft Windows. The software that runs many businesses was written to work only on Microsoft Windows. Users who are making the transition to Linux are more comfortable when the old familiar Microsoft operating system is there, just in case.

There are alternatives. One option is to use two separate computers. You can use the software created as part of the WINE (Wine Is Not an Emulator) project, which allows you to use some Microsoft applications on Linux. You could try related software, such as that offered by Xandros, Lindows, or CodeWeavers (CrossOver Office). You could even install Microsoft Windows inside a Linux third-party proprietary virtual machine application, such as VMware or Win4Lin.

However, the most commonly used option is still a dual-boot configuration. In other words, you can set up two different operating systems on the same computer. For example, Figure 2.1 shows the standard GRUB menu, configured to start either Red Hat Enterprise Linux or a Microsoft Windows server operating system.

FIGURE 2.1
A dual-boot configuration

You can set up a dual-boot with operating systems on separate physical hard drives. Alternatively, you can reconfigure the available free space on an existing hard drive. In this second case, carefully follow the procedures described in this chapter.

Whatever you do, start by backing up your data. Mistakes happen, and you want to be able to recover from a disaster.

The Easy Way: A New Hard Drive

In this section, we describe the easiest way to install Linux on an existing computer. Your BIOS should detect the second hard drive automatically. When it does, you know that the Red Hat installation program, Anaconda, should also detect that drive automatically.

As long as you limit the changes to the new empty drive, the risks are minimal. You can configure and format partitions with fewer risks to your Microsoft Windows data on the existing hard drive.

WARNING *The default settings for a Red Hat Enterprise Linux server installation will remove all data on all hard drives, even if this includes Microsoft Windows.*

As of this writing, Red Hat Enterprise Linux can be installed directly only on a regular IDE or SCSI hard drive. While you can use hard drives connected through USB, IEEE 1394, or parallel ports to store Linux directories, Anaconda may not support the installation of Red Hat Enterprise Linux on these drives.

NOTE *IEEE 1394 systems are also known by their proprietary names, FireWire (Apple's trademark) and iLink (Sony's trademark).*

WARNING *One more reason to add a new hard drive: Anaconda will not install Red Hat Enterprise Linux on hard drives with bad blocks.*

The Cheaper Way: An Existing Hard Drive

Not everyone can get a second hard drive. Even if you do, you may not have room inside your computer for that drive. Many people who want to set up Linux and Microsoft Windows in a dual-boot configuration will need to use the free space on an existing hard drive.The first Red Hat Enterprise Linux installation CD includes FIPS, which can help you split FAT formatted partitions.

NOTE *FIPS, the First Interactive Partition Splitter, can split only primary partitions. And as of this writing, it cannot split partitions formatted to Microsoft's NTFS filesystem. There are alternatives not included with Red Hat Enterprise Linux, described online at* `mlf.linux.rulez.org/mlf/ezaz/ntfsresize.html`.

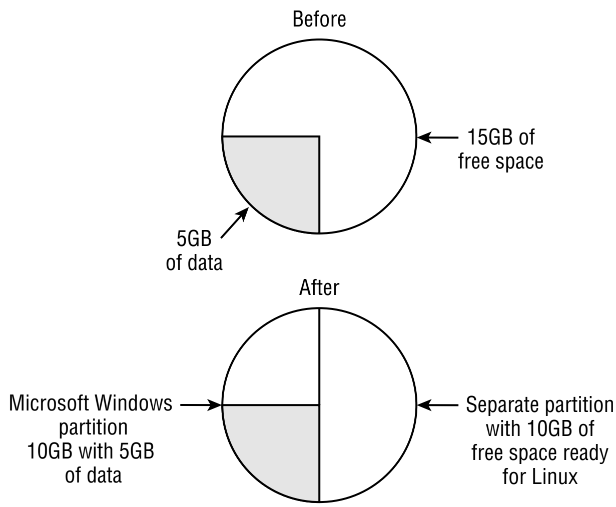
If you want to install Red Hat Enterprise Linux on the available free space on your hard drive, follow this basic procedure. Keep in mind that deviations can put your current data at risk. These are basic steps; the next section describes the dual-boot configuration process in detail.

1. Make room on your Microsoft Windows physical hard drive. In most cases, you can use Microsoft's Disk Defragmenter program for this purpose. To learn about the space you need for Linux, see Chapter 3.
2. Split the partition with your Microsoft Windows data. If you have a FAT or VFAT formatted partition, this is possible with `FIPS.EXE`. If you prefer, third-party tools are also available. Be sure to leave enough room for Microsoft Windows virtual memory.
   - **A.** Alternatively, if you want to split an NTFS partition, download one of the alternatives described earlier; there are commands available with the `ntfsresize` package.
3. If desired, use `fdisk` to organize the new partitions. You can also do this when you run Anaconda. If you use Microsoft's `FDISK.EXE` program, you'll be able to create only one primary partition.
   - **A.** Alternatively, with Windows NT/2000/XP/2003, you may want to use the Microsoft Disk Administrator to organize your new partitions.

If you want to use the free space on an existing hard drive, some planning is required. Make sure the free space is sufficient for all the Red Hat Enterprise Linux programs and packages that you want to install. Remember to include additional free space for your users' data and for any applications that you might install at a later date. We discuss space requirements in more detail in Chapter 3. And make sure that there is sufficient space for your Microsoft Windows operating system and its virtual memory requirements.

As an example, look at Figure 2.2. This is a view of a 20GB hard disk. You have 5GB of files for Microsoft Windows. You could easily split this hard drive into two partitions of 10GB each. The first partition would include enough free space for the Microsoft Windows files and virtual memory. The second partition would include enough room for installing everything from the Red Hat Enterprise Linux installation CDs.

FIGURE 2.2
Hard disk dual-boot scenario



TIP *Microsoft Windows requires significant free space for virtual memory. Linux does not require this kind of free space because Linux virtual memory is normally contained in a separate swap partition. In my experience, a Microsoft Windows partition doesn't work well if it's more than 60 percent full of files. But this is just a guideline; this is not a book about optimizing Microsoft Windows. If you want more information on this topic, Sybex has some excellent books in this series, including* Mastering Windows 98, Second Edition, Mastering Windows 2000 Professional, Second Edition, Mastering Windows XP Home, Second Edition*, and* Mastering Windows XP Professional, Second Edition*.*

NOTE *Some Linux users prefer a different utility,* `parted`*. You can use this GNU program to add, delete, resize, and format partitions. It doesn't yet work with NTFS partitions as of this writing. I'm hopeful that eventually it will incorporate the functionality of* `mkfs`*,* `FIPS.EXE`*, and* `fdisk`*. As of this writing, it makes changes to disk immediately, and therefore I consider it a riskier tool than* `fdisk`*. One advantage is that, like Partition Magic or System Commander, it can resize existing partitions. More information on* `parted` *is available from the GNU project at* `www.gnu.org/software/parted`*.*

Step-by-Step Procedure for VFAT Partitions

With the "big picture" in mind, you're ready to go through the step-by-step procedure of preparing your hard drive for Linux. This section assumes you're splitting a partition formatted to one of the older Microsoft format systems. These formats—FAT, FAT16, FAT32, or VFAT—are all known as VFAT on a Linux computer. This section also assumes you want to install Linux on the same physical hard drive where you already have Microsoft Windows installed.

Warning *This section uses FIPS. Use it at your own risk. FIPS explicitly comes with "ABSOLUTELY NO WARRANTY." I've used it frequently without problems; however, it is fairly easy to accidentally destroy your data with FIPS.*

This section assumes your hard drive is organized as only one partition, with all space allocated to the Windows C: drive. Alternatively, you could use these steps if the other drives don't provide enough room.

If you already have a hard disk with two or more partitions, you'll probably see this in Microsoft Windows as at least a C: and a D: drive. If you can move all of your files to the C: drive and still have enough room for Windows virtual memory, you can skip this process. Just make a note of the size of each of these drives to help you identify them during the Linux installation process.

To prepare your hard disk for Linux, follow these steps:

1. Find the capacity of your hard disk and the amount of space occupied by existing files. Determine the amount of room you want to allocate to Microsoft Windows and Red Hat Enterprise Linux.
2. Defragment your hard disk. Use the Disk Defragmenter, which is typically available from the Windows Start menu in the `Programs/Accessories/System Tools` folder. The exact steps and location vary depending on your version of Microsoft Windows.
3. Prepare a partition splitter. If you want to use `FIPS.EXE`, copy it, along with `RESTORRB.EXE` and `ERRORS.TXT`, to a Microsoft Windows or MS-DOS boot disk. Alternatively, you can use the boot disk that comes with a third-party partition splitter such as Partition Magic or System Commander. In the remaining steps, I assume that you're using FIPS.

Note *You can create MS-DOS boot disks on a Microsoft Windows computer with a floppy drive from downloads available at* `www.bootdisk.com`. *I prefer the Microsoft Windows 98 boot disk.*

4. Reboot your computer with the boot floppy. When you see the DOS A:\ prompt, run the `FIPS` command.
5. After you see the warning about not using FIPS in a multitasking environment, you see directions to "Press any key" to continue.

   If you have more than one hard drive, you're asked to choose; they're listed in boot order, similar to what's shown here. Drive 1 should be the first IDE or SCSI hard drive on your computer. Select a drive.

   ```
   Which Drive (1=0x80/2=0x81)
   ```

6. Next, you see a partition table (see Figure 2.3), listing the four primary partitions. If all four primary partitions are used, FIPS will fail, because it can split only primary partitions. If you have more than one partition that you can split, you're asked to select it, by number.

FIGURE 2.3
The FIPS partition table

```
If you use OS/2 or a disk compressor, read the relevant sections in FIPS.DOC.

FIPS comes with ABSOLUTELY NO WARRANTY, see file COPYING for details
This is free software, and you are welcome to redistribute it
under certain conditions; again see file COPYING for details.

Press any Key
Which Drive (1=0x80/2=0x81)? 2

Partition table:

     |        |      Start      |      |       End       | Start   |Number of|
Part.|bootable|Head Cyl. Sector|System|Head Cyl. Sector| Sector  |Sectors  |  MB
-----+--------+----------------+------+----------------+---------+---------+----
1    |      no |   1     0      1|   0Ch| 254 1304     63|       63| 20964762|10236

2    |      no |   0     0      0|   00h|   0    0      0|        0|        0|   0
3    |      no |   0     0      0|   00h|   0    0      0|        0|        0|   0
4    |      no |   0     0      0|   00h|   0    0      0|        0|        0|   0

Checking root sector ... OK

Press any Key
_
```

7. If you see the following message, you have to select from among the available primary partitions. Make your selection and continue. (If you select an extended partition, FIPS won't be able to handle it and will abort.)

Which Partition do you want to split (1/2/3)?

1. Your selected partition is scanned. You're shown basic information about the partition, and then you're asked whether you want to write backup copies of the boot and root sectors to a bootable floppy disk. This is an excellent idea. Answer **YES** to both questions. You see a message similar to `Writing file a:\rootboot.000`. Make a note of this file. If you have a problem, you can restore the original partition table by using the `RESTORRB.EXE` command.

2. Now you can define how you're going to split the partition. Using the arrow keys, you can change the size of the existing and new partitions. Make a note of the size of the new partition.

```
Old partition          Cylinder        New Partition
  4016.2 MB               512             6220.5 MB
```

3. When you're ready, press Enter to confirm the two new partitions. The old partition should contain the existing data. Next, FIPS tests the space to be occupied by the new partition. If it's empty, FIPS presents you with a new partition table similar to Figure 2.3. Next you must decide whether you want "…to continue or re-edit the partition table (c/r)?" If you press R, return to step 6. If you like your changes, press C to continue.

4. Finally, you're asked whether "…you want to proceed (y/n)?" to write the new partition scheme to disk.

5. Once the new partition scheme is written, you're ready to install Linux. The new partition should show up during the Red Hat Enterprise Linux installation process. If all goes well, it should show up as empty, and it should be the size you created with FIPS.

More information is available on FIPS from its website at `www.igd.fhg.de/~aschaefe/fips`.

Generic Procedure for NTFS Partitions

Before you start this process, back up the data on the NTFS partition that you want to split. There are a substantial number of boot CDs and floppies designed to boot Linux with NTFS partition management tools, and none of them are part of Red Hat Enterprise Linux as of this writing. One resource with several options is the `ntfsresize FAQ` website at `mlf.linux.rulez.org/mlf/ezaz/ntfsresize.html`. Because of the variety, a step-by-step procedure is not possible; however, these methods share a few characteristics:

- They involve a boot floppy or CD.
- They include some compressed miniaturized form of Linux that you can load on your computer.
- They provide a way to load the `ntfsresize` package on your system.

Detailed directions vary depending on the boot CD or floppy you select. You can even load it during the Red Hat Enterprise Linux installation process on the second virtual console described in Chapters 3 and 4.

Once loaded on your system, follow these basic steps:

1. Check your current partition layout with the `fdisk -l` command. For example, this command on my Windows NT 4 computer leads to the following output:

```
Disk /dev/hda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot    Start     End      Blocks   Id  System
/dev/hda1          1    1305    10482831    7  HPFS/NTFS
```

2. Now that you see the NTFS partition on your computer, use the `ntfsresize` command to find the available space on the partition of your choice. The location of the command may vary. Based on step 1, that would lead to the following command:

```
# /usr/sbin/ntfsresize --info /dev/hda1
```

3. The last line from the previous command should tell you how much you could shrink the current NTFS partition. Before you continue, you should test how you want to resize the partition. The following command tests what would happen if you shrank the partition to 6,000MB; the `--no-action` option prevents any actual resizing. If you see an error message, either select a different size or return to Microsoft Windows and try defragmenting the partition.

```
# /usr/sbin/ntfsresize --no-action --size 6000M /dev/hda1
```

4. Assuming you have no errors, you're ready to proceed with resizing. Remember, the `ntfsresize` tools are fairly new, so you really should back up any data on this partition before proceeding.

```
# /usr/sbin/ntfsresize --size 6000M /dev/hda1
```

5. Finally, you'll need to use `fdisk` to re-create the partition with the new partition size. You'll need to make sure you use the same starting disk cylinder, set the same NTFS partition type, make sure the partition you create is bigger than the resized partition, and set the bootable flag, if it was set before on this partition. For detailed information on `fdisk`, see Chapter 7.

WARNING *If you forget to create a new NTFS partition with* `fdisk`*, you won't get the room that you set up with the* `ntfsresize` *command. If you forget to set the NTFS partition type in* `fdisk`*, you may lose the data in your original NTFS drive.*

Why Worry about Hardware?

The community of developers that supports Linux has done an excellent job creating drivers for an overwhelming majority of workstation and server hardware. Many—perhaps even most—new components get Linux drivers within months of their release. Many hardware manufacturers, in fact, include Linux drivers with their hardware or make them available for download from their websites. With the advances in Linux plug and play, most hardware is now detected and configured automatically. So in many cases, you don't have to worry about hardware when you install Red Hat Enterprise Linux on your computer.

However, you can have problems. If you're planning to install Linux on a group of computers, hardware problems can be expensive. Not all hardware is built for Linux—or for Microsoft Windows 2003, for that matter. And not all hardware has Linux drivers.

Hardware Problems Can Be Expensive

It's true that the cost of hardware tends to fall over time. However, when you're planning for a group of computers, the cost of replacing every network card quickly adds up, not only in hardware, but also in the labor required for each computer.

Some components are more expensive than others. If you make a mistake with your video configuration, you could easily blow the circuits associated with your monitor. And if that monitor is your laptop display, the cost can be frightening. Therefore, you should at least record the specifications for your video card and monitor.

If you make a mistake while configuring a video adapter, you could make it send signals that exceed the capability of your monitor. This is true on Linux as well as Microsoft Windows computers.

NOTE *In most cases, modern monitors just tell you that you've made a mistake.*

When video adapters send signals to monitors, they send them at specific frequencies and refresh rates. Monitors have limits on the frequencies and refresh rates that they can handle. The results could burn out circuits on your monitor. While some monitors have protective circuits built in, why take the risk?

Not All Hardware Is Built for Linux

Some manufacturers release the source code for their hardware. Some of this code is even released under the General Public License (GPL). This makes it easy for a Linux developer to design a driver for that hardware component.

However, not all hardware is built for Linux. For example, a group of modems and printers, Winmodems and Winprinters, were explicitly designed for Microsoft Windows. They explicitly use Microsoft Windows driver libraries to function. Since Microsoft doesn't release the source code for its driver libraries, this makes it difficult for Linux developers to create drivers. Strangely enough, because of the changes in Microsoft Windows XP/2003, many Winmodems and Winprinters often don't work on these latest Microsoft operating systems.

TIP *A number of Linux books suggest that you avoid Winmodems at all costs. That may no longer be necessary. I have Winmodems that Linux recognizes on both my laptop and desktop computers.*

Sometimes Linux developers haven't had the time to create drivers for the latest components. As of this writing, Linux drivers are incomplete for three types of components: USB, IEEE 1394, and IEEE 802.11 wireless systems. While Linux support for USB 1.*x* components is fairly good, USB 2.0 requires a kernel that supports the Enhanced Host Controller Interface (EHCI), which is still experimental for the kernel that's supplied with Red Hat Linux 9.

Linux support for some IEEE 1394 equipment is available as experimental drivers. Linux support for regular wireless networking (IEEE 802.11b) is good; drivers for IEEE 802.11a–11g are just being proven as of this writing (I've installed one on my laptop computer). Later in this chapter, in the "Questionable Hardware" section, you can find the home pages for those who are developing these cutting-edge drivers.

TIP *Starting with version 8.0, Red Hat Linux distributions can no longer be installed on computers with 386- and 486-level CPUs.*

Red Hat Enterprise Linux Supports Many Architectures

You can install Red Hat Enterprise Linux 3 on a wide variety of computers, not just 32-bit PCs with Intel-compatible CPUs. However, when you're installing this operating system on a less-common platform, certified hardware becomes more important. It's not possible to detail what you can do with all supported architectures. I've summarized these architectures here:

x86 The baseline Intel 32-bit architecture forms the foundation of personal computers and entry-level servers today. This category includes computers with compatible CPUs, including those made by AMD and other manufacturers. Strangely enough, while generic x86 software packages are in "i386" format, Red Hat Enterprise Linux 3 can't be installed on computers with Intel 386- or even 486-level CPUs.

Depending on whether you get the WS, ES, or AS versions, you can set it up on computers of up at least 2 and up to 16 CPUs. Different kernels are available that are optimized for different x86 CPUs and memory levels. Red Hat supports computers with between 256MB and 64GB of RAM.

Itanium The Itanium CPU is Intel's 64-bit CPU. Red Hat Enterprise Linux 3 supports the Itanium2 architecture with one to eight CPUs. It requires a system with between 512MB and 32GB of RAM.

It includes the Extensible Firmware Interface, also known as the EFI shell. It's a command-line interface associated with Itanium systems that you can use to start the installation of Red Hat Enterprise Linux 3.

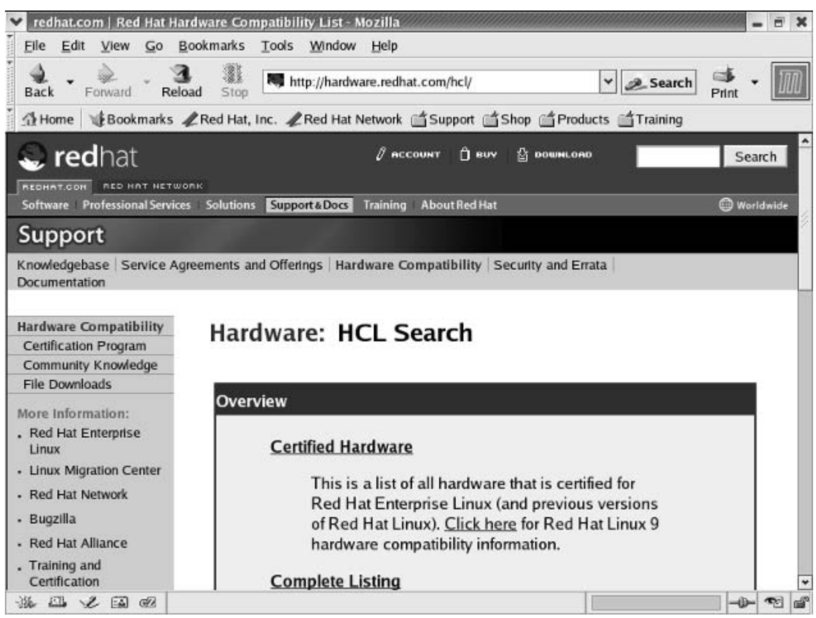
AMD64 You can also install Red Hat Enterprise Linux 3 on an AMD 64-bit architecture. Red Hat supports this architecture with one to four CPUs. It requires a system with between 512MB and 16GB of RAM. The basic installation interface is the same as the x86.

IBM architectures You can install Red Hat Enterprise Linux 3 Advanced Server on all IBM-based server platforms. For more information on the available choices and setting up partitions on these IBM servers, refer to `www-1.ibm.com/servers/eserver/linux`.

Finding Compatible Hardware

On its website, Red Hat includes the latest available information on compatible hardware. Visit the hardware compatibility section of its site, currently available at `hardware.redhat.com/hcl`, as shown in Figure 2.4. Linux-compatible hardware is often organized in what's known as a hardware compatibility list (HCL).

FIGURE 2.4
Red Hat's hardware compatibility list



Red Hat has tested hardware on a number of PCs. However, the company also relies on the work of other Linux developers. Red Hat classifies hardware in one of the four categories described in Table 2.2.

TABLE 2.2: RED HAT HARDWARE COMPATIBILITY CATEGORIES

CATEGORY	DESCRIPTION
Certified	Hardware that has been officially tested by Red Hat through its official certification program and that's known to work with Linux.
Compatible	Hardware that has been reviewed by Red Hat personnel, outside the official certification program.
Community knowledge	Hardware that has been found by others to be compatible with Linux. While Red Hat may include drivers for such hardware as part of the installation CDs, it is not supported by Red Hat, Inc.
Not supported	Hardware that has been officially tested by Red Hat through its official certification program and that's known to *not* work with Linux.

In the following sections, we describe examples of each category of hardware. Red Hat also provides at least 30 days of installation support, even for Red Hat Professional Workstation. However, while Red Hat support is excellent, it may not be able to solve every problem you may have.

Red Hat Enterprise Linux–Certified Hardware

Certified hardware has been officially tested by Red Hat. Generally (with some exceptions), you'll find entire systems, such as IBM-branded servers, listed as Red Hat–certified hardware, but not single components.

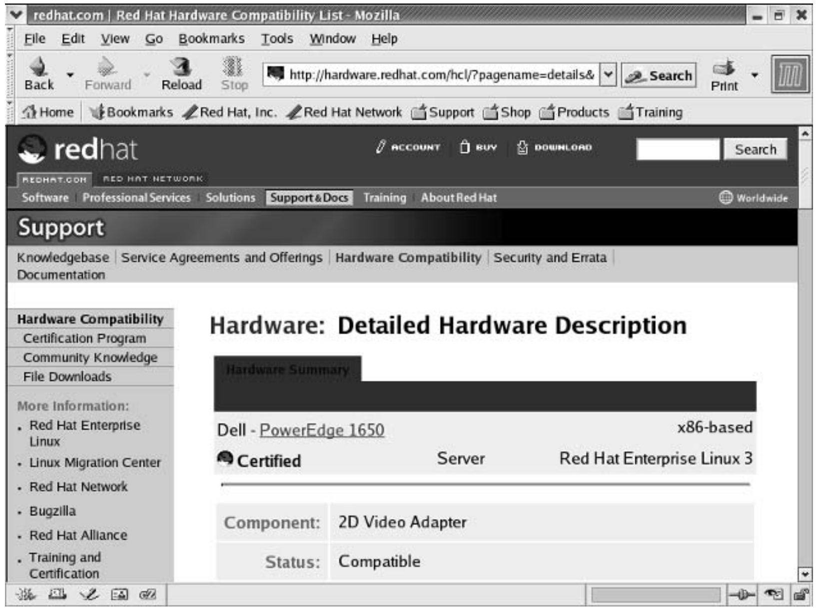
On the Red Hat HCL web page, you can click the Hardware Compatibility List link to navigate to a search engine for the Red Hat HCL. It's a straightforward search engine. Once you've found hardware, you can find Red Hat's review of the component. For example, Figure 2.5 shows Red Hat's review of a Dell PowerEdge server.

From this web page, you can see that the Dell PowerEdge 1650 server is certified for Red Hat Enterprise Linux 3, and the list breaks the certification down to key hardware components. It includes the video adapter, CD-ROM drive, controller card, CPU, hard drive, floppy drive, RAM, and network interface card. All drivers are included on the Red Hat installation CDs and are rated as easy to install.

Compatible Hardware

There is a subtle difference between certified and compatible hardware. Certified hardware generally consists of entire systems; compatible hardware includes individual components such as CPUs, hard drives, graphics adapters, and network cards. It's difficult to test every possible combination of components; unknown interactions can affect compatibility with any operating system.

FIGURE 2.5
A Red Hat review of a server



Red Hat provides limited support to licensed users of Red Hat Enterprise Linux for "compatible hardware." It's easy to find a list of compatible hardware on the Red Hat HCL. Navigate to `http://hardware.redhat.com/hcl`. At the top of the page, you'll see four tabs that can help you search through the Red Hat HCL.

I've performed an "Advanced Search" for Desktop/Workstation computers that are known to be compatible with Red Hat Enterprise Linux 3; the results are shown in Figure 2.6. As you can see, the "Certified" list is quite limited, but in reality, you can safely install this operating system on a wide variety of different desktop and workstation computers

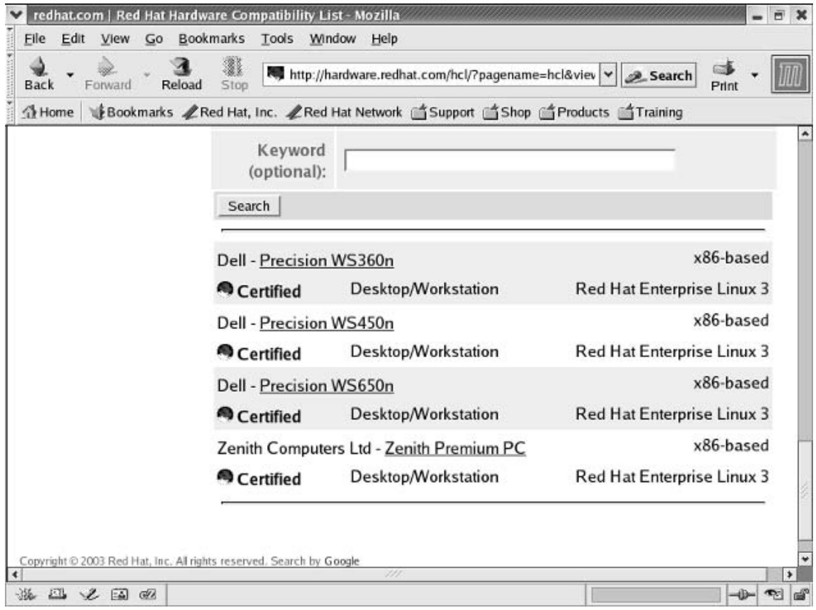
NOTE *Web links may change by the time you read this book. If the link doesn't work, you'll have to use your own insight on the Internet to find the data you need. You should be able to find it in a support area of the site.*

Questionable Hardware

There are several categories of hardware where Linux support is less than ideal. Yet Linux developers have made progress in a number of areas. For example, it is no longer necessary to avoid all Winmodems. If you have questions about your hardware, a good place to start is the Hardware Compatibility HOWTO of the Linux Documentation Project, currently available at `www.tldp.org/HOWTO/Hardware-HOWTO`.

We've listed several different categories of questionable hardware, along with resources that can help. If you can't find a driver for some component on your PC, look through the associated websites. An enterprising Linux developer may have the driver or solution for you.

FIGURE 2.6
Compatible video hardware



TIP *When you look for drivers, you may not need a perfect match. For example, you may be able to configure a video card using an older driver from the same manufacturer.*

Cameras Digital camera manufacturers generally use proprietary programs and interfaces. Despite these limits, the gPhoto2 developers have developed software that works with hundreds of digital cameras (see `www.gphoto.org`).

FireWire/iLink/IEEE 1394 FireWire and iLink are trade names for the IEEE 1394 standard. It supports high-speed data transfer for external devices such as hard disks and video cameras. While kernel support for these devices is officially still "experimental," help is available through `www.linux1394.org`.

Graphics cards Red Hat Enterprise Linux works fine with almost all graphics cards, at least in VESA (Video Electronics Standards Association) mode, as described in Chapter 3. But developers are improving drivers all the time. Assuming you use the default XFree86 Server, you may be able to find a driver update for your card through the XFree86 project at `www.xfree86.org`.

Laptops Red Hat Enterprise Linux 3 works fine on most laptop computers. I've installed it with minor tweaks on my laptop computer. However, there are risks, because laptop manufacturers use a considerable amount of proprietary software. The Linux on Laptops web page at `www.linux-laptop.net` offers the experience of a number of users on different laptop computers. It helped me install this operating system with a minimum of problems. The Linux-Mobile-Guide provides

detailed information on configuring laptop computers and other mobile devices at `http://tuxmobil.org/howtos.html`.

Network cards Red Hat Enterprise Linux works well with most standard network cards. But as network speeds increase, new network cards are under development; you may not find the latest driver for all Gigabit or 10 Gigabit Ethernet network cards on the Red Hat CDs. Development work on the latest Linux network card drivers is sponsored by Scyld Computing, at `www.scyld.com/network`.

Printers The so-called Winprinter can be as difficult to configure as the Winmodem, and new printers with more features are being released at an astonishing rate. The developers at `www.linux-printing.org` have done amazing work developing new print drivers and configuration files.

Scanners The Scanner Access Now Easy (SANE) home page provides tips and tricks for configuring regular and USB scanners for Linux. Currently, the home page for SANE development is at `www.sane-project.org`.

Sound cards Sound cards can be difficult to configure in Linux. For example, some cards need multiple DMA channels; others can be configured to emulate one of the Sound Blaster cards. The latest information in Linux sound card support is available from the Advanced Linux Sound Architecture (ALSA) project at `www.alsa-project.org`.

USB While Red Hat Enterprise Linux can detect the basic USB keyboard and mouse during the installation process, Linux support for USB devices is currently less than ideal. But as this industry moves toward converting external devices to USB and IEEE 1394 standards, Linux developers will be creating new drivers for every type of external hardware. As of this writing, support for IEEE 1394 and USB 2.0 standard high-speed equipment is officially still experimental. The latest information on Linux support for USB is available from the Linux USB Project at `www.linux-usb.org`.

Winmodems As described earlier, Winmodems depend on Microsoft Windows driver libraries to support their functionality. However, the people behind the Linmodem project have developed Linux drivers that work seamlessly with many Winmodems. Many of their drivers are incorporated into Red Hat Enterprise Linux 3. Many Winmodems are now detected automatically through the Linux plug-and-play system. But not all Winmodems work in Linux. For the latest status, see `www.linmodems.org`.

Community Knowledge Hardware

The Linux operating system is based on a collective effort of developers from around the world. People in the Linux community have organized themselves into a number of groups. As described earlier, many of these groups are dedicated to creating and updating drivers for specific types of hardware. Their progress is documented at their websites and in mailing lists.

When you want the latest community knowledge about Linux hardware, there are four ways to direct your research. The Linux Hardware HOWTO provides an overall view of Linux hardware compatibility. However, it may not include the latest hardware information. More data is available at the websites for many hardware-specific Linux support groups, as described in the previous section. Many of these groups have open mailing lists, where developers exchange information on their latest

work. Finally, users ask questions about hardware all the time on the Internet newsgroups. A searchable newsgroup database is available at `http://groups.google.com`.

NOTE *Before asking a question on a mailing list or newsgroup, do your research first. Many Linux developers have jobs and don't have time to give you answers that can already be found in documentation, such as the LDP HOWTOs at* `www.tldp.org`. *In fact, many will show their annoyance if you waste their time. Before you ask a question on a newsgroup or mailing list, check the documentation available on the subject. Search the newsgroups or mailing list database message archives to see if your question has been answered before.*

Creating a Hardware Checklist

Ideally, you should collect information on every hardware component in your computer. This section provides a checklist on the information that you need. Once you've identified your hardware, you can check the Red Hat and other websites for the drivers and configuration tips that you may need.

At a minimum, you should get the specifications for your graphics card and monitor before installing Red HatEnterprise Linux 3. Once Linux is installed, test each hardware component. Make a list of those components that are hard to configure or that do not work to your satisfaction. Detected components are normally configured in the `/proc` directory, as described in Chapter 11. The next time you install Red Hat Enterprise Linux, you'll be ready with the drivers and configuration commands that you need. This is a good approach if you're installing Red Hat Enterprise Linux on a group of computers.

In the following sections, you'll learn about the information that you should collect on each hardware component. Then, you'll find how to associate each component with a specific driver. Finally, we provide a table where you can fill in the blanks with the data you need.

Collecting Information

Before starting to install Red Hat Enterprise Linux on your computer, you should keep in mind a few basic things. You don't absolutely need to know everything about every hardware component; most are automatically detected during the installation process. Review the list of priority hardware in Table 2.3—which applies only to x86 systems. For more information, see `www.redhat.com/software/rhel/configuration/`.

TABLE 2.3: PRIORITY HARDWARE

COMPONENT	REQUIRED INFORMATION
CPU	Red Hat Enterprise Linux 3 requires at least a 300MHz Pentium-level CPU.
RAM	Red Hat supports configurations with at least 256MB of RAM (if your computer shares RAM for your video hardware, you may need a bit more). However, I've installed this operating system on a VMWare workstation with as little as 96MB of RAM.
Graphics card	You need to know a bit about your video card. The Linux XFree86 server packages include a database that can configure your card based on the make and model. If Linux doesn't recognize the card, you should be able to configure it separately knowing the video RAM and available vertical and horizontal refresh rates.

Table 2.3: Priority Hardware *(continued)*

Component	Required Information
Monitor	You should know the capabilities of the monitor: its resolution, as well as its vertical and horizontal refresh rates. If the graphics card can put out refresh signals greater than the monitor's capacity, be careful; the wrong settings can burn out your monitor.

In other words, you should know the make, model, and specifications of at least the priority hardware components on your computer.

Collecting Drivers

Drivers for most hardware components are already included with the Red Hat Enterprise Linux installation CDs. Most drivers are automatically configured during the Linux installation process.

But Red Hat Enterprise Linux 3 does not include drivers for all hardware. No Microsoft operating system includes drivers for all hardware. There are two basic ways to collect additional drivers. One is based on community knowledge, as discussed earlier. The other is based on drivers created by hardware manufacturers.

Many hardware manufacturers are friendly to Linux. Remember, IBM has invested more than a billion U.S. dollars in Linux development just in 2001. A lot of manufacturers have followed their lead and provided Linux drivers for their hardware. Many Linux drivers are downloadable from manufacturer websites. Typically, documentation and instructions are available from the same sites.

Once drivers are available, they can be installed with commands such as `insmod`. You can make sure the drivers are installed the next time Linux starts with the right commands in `/etc/modules.conf`. More information on this process is available in Chapter 11.

Hardware Checklist

For your convenience, this section includes the hardware information that you should collect for your PC. This is more important if you have a group of PCs with similar configurations so that you avoid potentially costly errors. Table 2.4 lists the hardware you need to detail.

You should make special note of any devices that don't conform to plug-and-play standards. You may need to reserve IRQ ports or I/O addresses in your BIOS for any such hardware.

Table 2.4: Hardware Checklist

Component	Detail
CPU type, speed	
RAM memory, in MB	
Keyboard, make, model	
Mouse, protocol, make, model, buttons	
Hard drive 1 size	

Table 2.4: Hardware Checklist *(continued)*

Component	Detail
Partitions and mount points, such as `/home` and `/dev/sda1`	
Hard drive 2 size	
Partitions and mount points, such as `/var` and `/dev/sdb1`	
Hard drive 3 size	
Partitions and mount points, such as `/usr` and `/dev/sdc1`	
Hard drive 4 size	
Partitions and mount points, such as `/boot` and `/dev/hda1`	
CD drive, type	
DVD drive, type	
SCSI adapter, make, model	
Network card, make, type, model, speed	
Network card 2, make, type, model, speed	
Telephone modem, make, model, speed	
Graphics card, memory, make, model, vertical and horizontal refresh rates	
Monitor, make, model, vertical and horizontal refresh rates	
Sound card, make, model, chipset	
USB device 1, make, model	
USB device 2, make, model	
USB device 3, make, model	
IEEE 1394 device 1, make, model	
IEEE 1394 device 2, make, model	

BIOS Tips

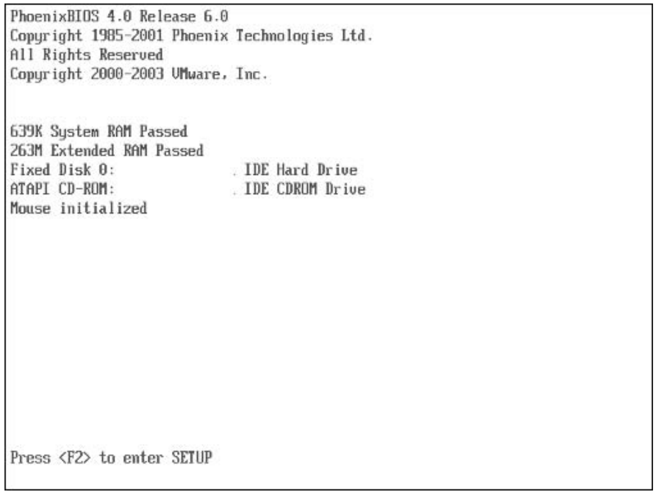
There are three things that you may be able to configure in your computer's BIOS. One is the boot order of your hard drives. Next is the boot sequence; for example, you can configure your BIOS to boot the Red Hat Enterprise Linux installation program from the appropriate CD. Finally, you may be able to reserve key communications channels, such as IRQ ports and I/O addresses.

NOTE *This section assumes you're working with a computer with an x86 or AMD64 architecture. Other architectures require different procedures, which you can find in the Red Hat documentation available at* `www.redhat.com/docs`*.*

A wide variety of BIOS menus are available. It's therefore not possible to provide specific directions on how to configure a BIOS. What you can configure depends on the BIOS menu and any upgrades you may have installed.

Normally, you can review your BIOS menu by pressing a key such as F1, F2, or Del on your keyboard just after the initial beeps on your computer. Sometimes, you'll see a menu such as Figure 2.7 during the boot process.

FIGURE 2.7
PC startup menu



Sometimes the menu is hidden, perhaps by a screen associated with your computer or motherboard manufacturer. Press F1, F2, or Del. If one of these commands doesn't start your BIOS menu, consult the documentation for your PC or motherboard. In the BIOS menu, you should see the detected IDE drives.

NOTE *With some Compaq and Acer computers, you'll need to press Ctrl+Alt+Esc to access the BIOS menu.*

Naturally, these instructions don't apply to Itanium 64-bit CPU systems with an EFI interface or IBM series servers.

IDE Hard Drives

On a standard PC, you may have up to four IDE drives. They may be hard drives or CD/DVD drives, and they should be detected as such in the BIOS menu.

If you've installed IDE drives and they're not detected in your BIOS, you may have a hardware problem. For more information on troubleshooting PC hardware installation, please refer to the *Complete PC Upgrade and Maintenance Guide, 15th Edition* (Sybex, 2004).

Standard PCs have two IDE adapters: a primary and a secondary. Each adapter can be connected to two different IDE drives: a master and a slave. Linux associates specific device files with these drives, as shown in Table 2.5.

TABLE 2.5: LINUX IDE DEVICE DRIVER DRIVE DEVICE FILES

Drive	Device File
Primary master	`/dev/hda`
Primary slave	`/dev/hdb`
Secondary master	`/dev/hdc`
Secondary slave	`/dev/hdd`

SCSI Hard Drives

There are several different types of SCSI standards. SCSI-1, SCSI-2, and SCSI-3 standards are associated with a maximum of 8 or 16 devices, with data transfer speeds of up to 80MBps. Each SCSI device has an ID, which specifies its priority on your PC.

SCSI hard drives can be installed internally or externally. Most newer BIOSes can detect SCSI drives at least as part of its boot sequence menu. On older PCs, you may need a SCSI BIOS.

NOTE *IEEE 1394 drives are technically SCSI drives without LUN numbers. As of this writing, you can't boot Linux from an IEEE 1394 drive.*

Boot Sequence

In your BIOS menu, you should see a Boot Sequence option, which allows you to specify the boot order. Your PC's BIOS looks at these drives in order for the `/boot` directory for the Linux startup files and kernel. You can configure your PC to look to any detected drive first. However, you need to set up your BIOS to look to a specific drive for the `/boot` directory.

If you have IDE drives connected to both the primary master and primary slave attach points, `/boot` must be installed on one of these drives (`/dev/hda` or `/dev/hdb`). This applies even if a CD/DVD is connected to one of these attach points. If you have two primary IDE drives, the Red Hat Enterprise Linux installation program in fact forces you to configure `/boot` on one of these drives.

If you have one primary IDE drive and one SCSI drive, `/boot` must be installed on one of these drives. The SCSI drive must have an ID of 0.

If you have no primary IDE drives and two or more SCSI drives, `/boot` must be installed on one of the first two SCSI drives, with an ID of 0 or 1.

Non-Plug-and-Play Hardware

While Linux can now detect most plug-and-play hardware, some legacy devices don't conform to plug-and-play standards. In many newer BIOS menus, you can reserve IRQ ports and I/O addresses for such hardware. For example, an older network card may require a standard port, such as IRQ 10, and a standard I/O address, such as 0x300. If you can reserve these locations, you can configure that network card appropriately after Linux is installed.

Post-Installation Hardware Configuration

Just because you've installed Red Hat Enterprise Linux doesn't mean that all hardware on your computer is playing well with this operating system. If you add or remove memory or CPUs, you can make sure the system is still supported with the `redhat-support-check` command. Hardware communicates with the kernel, using settings as described in the `/proc` directory. The Red Hat Hardware Browser can help you view detected hardware in the GUI.

There are several other tools that can help you configure certain hardware components after installation. The Red Hat `Keyboard`, `Mouse`, and Sound `Card` configuration tools are straightforward. Other hardware changes are normally detected during the boot process with the `kudzu` configuration tool.

Quick Checks with *redhat-support-check*

The `redhat-support-check` tool is straightforward. When run it at the command-line interface, it checks your current CPU and RAM configuration to make sure it's still supported. It checks your system against the data listed in your `/var/lib/supportinfo` file. If your system is supported, you'll see no output.

/proc directory

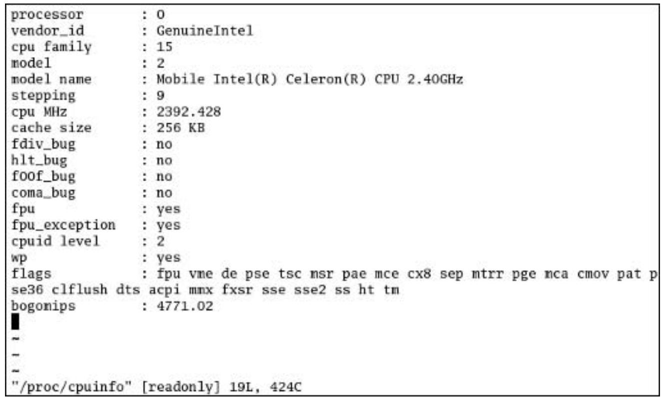
Linux makes it easy to see how the Linux kernel views your hardware. Just look in the `/proc` directory. As shown in Table 2.6, various files in `/proc` can give you additional information on the hardware that's connected to a Red Hat Enterprise Linux computer.

Table 2.6: Selected Hardware files in /proc

File	Description
`apm`	Advanced power management battery status
`cpuinfo`	Detected CPUs
`dma`	Assigned DMAs
`ide`	Directory specifying attached IDE devices
`interrupts`	Assigned IRQs
`ioports`	Assigned I/O addresses
`modules`	Installed driver modules; same as `lsmod` output
`partitions`	Basic partition information
`pci`	Detected PCI devices
`scsi`	Directory specifying attached SCSI devices

The information is quite detailed. For example, look at the `/proc/cpuinfo` file in Figure 2.8. Not only does it show the rated and the effective speed of the CPU, but it also shows the cache size, another measure of the CPU. You'll see how this helps in Chapter 12.

FIGURE 2.8
Kernel information on the CPU



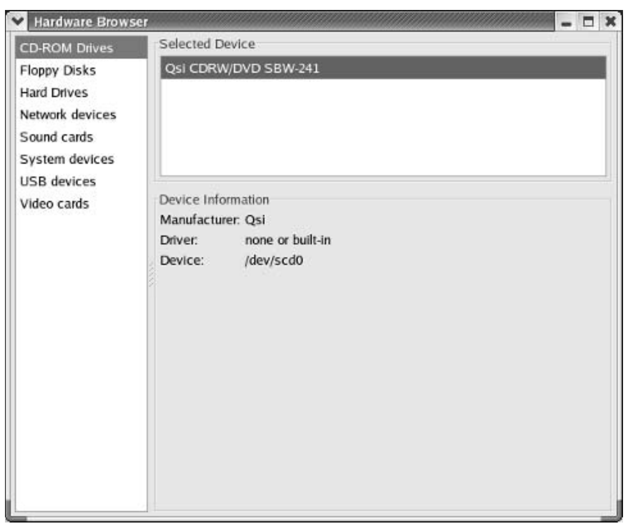
```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Mobile Intel(R) Celeron(R) CPU 2.40GHz
stepping        : 9
cpu MHz         : 2392.428
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 sep mtrr pge mca cmov pat p
se36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
bogomips        : 4771.02

~
~
~
"/proc/cpuinfo" [readonly] 19L, 424C
```

The Red Hat Hardware Browser

The Red Hat Hardware Browser takes some of the information from the `/proc` directory and collects it in a more readable format. For example, Figure 2.9 illustrates the detected CD-RW/DVD drive, as documented in the `/proc/scsi/scsi` configuration file.

The Hard Drives option may be especially interesting, as it illustrates detected partitions and filesystems in a format similar to the Disk Druid tool that you'll see during the installation process.

FIGURE 2.9
The Hardware Browser



The Red Hat Keyboard Tool

You can configure keyboards that correspond to a wide variety of languages and dialects or systems. You use the Red Hat keyboard utility to set the keyboard that most closely corresponds to your system.

Start this tool by selecting Main Menu ➢ System Settings ➢ Keyboard, or run the `redhat-config-keyboard` command from a GUI command-line interface. This opens the Keyboard window, shown in Figure 2.10.

FIGURE 2.10
Selecting a keyboard

As you can see in the figure, there can be more than one keyboard for different languages and national locations. If necessary, select the keyboard that most closely fits your hardware and click OK. Changes are reflected in `/etc/sysconfig/keyboard`.

TIP *If you run* `redhat-config-keyboard` *from a text-mode console, this utility will give you the same choices (in a different order) in a blue text-mode, low-graphics screen.*

The Red Hat Mouse Configuration Tool

You can configure many kinds of pointing devices. The most common pointing device today is a mouse; in fact, the terms *pointing device* and *mouse* are used interchangeably during the configuration process. To configure the settings for your default pointing device, start the Red Hat mouse configuration utility.

Start this utility by selecting Main Menu ➢ System Settings ➢ Mouse, or run the `redhat-config-mouse` command from a GUI command-line interface. This opens the Mouse Configuration window, shown in Figure 2.11.

The default is based on your current `/etc/sysconfig/mouse` file. Any changes you make are written there. If you have a two-button mouse, you may want to activate the Emulate 3 Button Click option. This allows your mouse to simulate the functionality of a middle mouse button when you press both the left and right buttons simultaneously. KDE uses the middle mouse button to open a pop-up menu of commands.

FIGURE 2.11
Configuring a mouse

If you have a mouse that's connected to a serial port, you'll be able to select the Serial Devices button. This opens a menu where you can select the actual serial device in use. If you've used this computer on a Microsoft operating system before, you should set your mouse to the associated Microsoft COM port per Table 2.7.

TABLE 2.7: MOUSE SERIAL DEVICES

Device File	Description
`/dev/ttyS0`	Corresponds to the Microsoft COM1 port
`/dev/ttyS1`	Corresponds to the Microsoft COM2 port
`/dev/ttyS2`	Corresponds to the Microsoft COM3 port
`/dev/ttyS3`	Corresponds to the Microsoft COM4 port

TIP *These serial devices also help you configure many telephone modems. If you've configured your modem in Microsoft Windows, pay attention to the COM port associated with the modem. If you run the* `ls -l /dev/modem` *command, you may see a link to the corresponding device file shown in Table 2.7. Other COM ports are available, especially if you have a Winmodem. In that case, you'll want to see* `www.linmodems.org` *for more information.*

If you've made a change and close `redhat-config-mouse`, Linux stops and restarts the mouse console.

TIP *You don't need a GUI to run* `redhat-config-mouse`*; if you're in a text console, Red Hat automatically starts a text-mode version of this utility.*

Sound Card Management (*redhat-config-soundcard*)

Red Hat Enterprise Linux lets you configure many kinds of sound cards. To set the settings for your default pointing device, start the Red Hat sound configuration utility.

Start this utility by selecting Main Menu ➤ System Settings ➤ Soundcard Detection, or run the `redhat-config-soundcard` command from a GUI command-line interface. This opens the Audio Devices window, shown in Figure 2.12.

FIGURE 2.12 Configuring a sound card

If `redhat-config-soundcard` detects a sound card on your system, you'll see the make and model of the card in the Audio Devices window. You can test the result by clicking the Play Test Sound button. Assuming you have a sound card, your drivers are detected, and your speakers are connected, you should hear a sound and get a confirmation window. Confirm the result. If Linux needs to install special kernel modules for your sound card, changes are written to `/etc/modules.conf`.

NOTE *Red Hat no longer includes the* `sndconfig-*` *RPM.*

Forcing Hardware Detection with *kudzu*

If Red Hat Enterprise Linux didn't detect additions or deletions to your hardware, try starting the Red Hat Hardware Discovery Utility, also known as `kudzu`. Sometimes `kudzu` can help you detect hardware changes.

It runs automatically during the boot process. If you've installed or removed a "hot-swap" component, you may need to run `kudzu` again. If it finds something new, it will offer to configure the hardware for you, with a screen similar to Figure 2.13.

Summary

Before you can install Red Hat Enterprise Linux, you need to prepare your hardware. You may have to prepare hard disk partitions on IDE and/or SCSI drives for Linux. Special preparations are required if you want to configure Linux and another operating system, such as Microsoft Windows, on the same computer.

If you already have Microsoft Windows installed, it's easiest to install Linux on a second, empty hard drive. The Red Hat Enterprise Linux installation program, Anaconda, should detect the new empty hard drive and configure partitions on this drive. If you don't have a second hard drive, all you need is sufficient room on the first drive. With the Microsoft Windows Disk Defragmenter, you can make room. Using the FIPS utility, you can split an existing partition into two. You can then install Linux in the free space of the newly created partition.

FIGURE 2.13
kudzu can detect a deleted network card.

Red Hat Enterprise Linux detects most current computer hardware, especially if you're installing on an x86 PC. Usually, there are no hardware concerns when installing Red Hat Enterprise Linux. But if you're planning to install Linux on a group of computers, problems can be expensive. Not all hardware is built for Linux. And some hardware, specifically related to the graphics system, can be put at risk during the installation process.

Red Hat can help you find compatible hardware. Red Hat classifies hardware in four categories: certified, compatible, community knowledge, and not compatible. Community knowledge hardware may require additional work; drivers, directions, and advice are available from a number of sources.

You should collect basic information at least on the CPU, RAM, and graphics system. Drivers are available from a number of sources, including those discussed as community knowledge, as well as from the websites of a number of hardware manufacturers. We provided a hardware checklist and table to help you collect data on the other components in your computer.

To prepare your x86 computer, you should also at least review the settings in your BIOS. The BIOS can help you configure IDE and SCSI hard drives. The Linux `/boot` directory should be installed on specific drives. The boot sequence should work with these drives. You can also reserve specific channels in many BIOS menus for non-plug-and-play legacy hardware. Other architectures such as Itanium 2 use different systems such as EFI.

There are a number of tasks that you can configure to monitor and change the hardware configuration after installation. Hardware kernel settings are normally stored in the `/proc` virtual directory; Red Hat includes a number of configuration tools that can also help.

In the next chapter, you'll install Red Hat Enterprise Linux, using various boot methods, from files on local Red Hat Enterprise Linux installation CDs. Once Linux is installed, you'll see how easy it is to register your computer for updates through your paid subscription to the Red Hat Network, as well as other options.

Chapter 3

Installing Linux on a Stand-Alone System

In this chapter, we'll look at the graphical Red Hat Enterprise Linux installation process, from the installation CDs, step by step. In most cases, all you need to do is set your computer to boot from the first Red Hat Enterprise Linux installation CD, restart your computer, and follow the prompts. You can also customize Red Hat Enterprise Linux to your specifications.

The Red Hat Enterprise Linux installation program is known as Anaconda. A very flexible program, it can accommodate separate boot disks, or, as you'll see in Chapter 4, it allows you to install over a network. If you're installing from CD, Anaconda includes a `mediacheck` option that inspects the integrity of your installation CDs. If it recognizes a previous installation of Red Hat Enterprise Linux on an x86 system, it supports upgrades.

This chapter focuses on the graphical Anaconda installation process from a CD, which per spec requires 256MB of RAM on your computer. If you want to install Red Hat Enterprise Linux over a network or use the Anaconda text-mode installation process, read Chapter 4.

Once the installation is complete, we will look at how you can diagnose typical installation problems. We'll then proceed with the first graphical and text login screens.

If you set Red Hat Enterprise Linux to log in graphically by default, you'll see the `firstboot` process the first time you restart your computer. It lets you synchronize your date and time with a network time server, search for a sound card, register your computer with the Red Hat Network, and install additional software.

This chapter covers the following topics:

- Starting with a boot disk
- Checking the installation CDs
- Installing Red Hat Enterprise Linux, step by step
- Running the Red Hat Setup Agent
- Troubleshooting the installation
- Logging in
- Upgrading Red Hat Enterprise Linux

Starting with a Boot Disk

In most cases, you can install Red Hat Enterprise Linux directly from your CD drive. All you should need to do is reconfigure the settings in your BIOS menu to boot directly from that drive, as described near the end of Chapter 2 (we briefly describe alternatives in the note that follows). However, there are situations where you need a boot disk:

- You're unable to set your BIOS to boot from your CD.
- Your CD is unable to read the boot files from the first Red Hat installation CD.
- You're installing Red Hat Enterprise Linux from another source, such as a remote computer through the network (covered in Chapter 4). If you can boot from a CD, you may prefer to create the fairly small `boot.iso` CD for this purpose.

If you need to install Red Hat Enterprise Linux from a boot floppy, you may need anywhere from one to four 1.44MB floppy disks, depending on your installation method and hardware.

You can create these floppies from `.img` files in the `/image` directory of the first Red Hat installation CD. The key files in this directory are summarized in Table 3.1.

Table 3.1: Red Hat Enterprise Linux Installation Images

Image File	Description
`bootdisk.img`	Standard boot disk for all local and network installations.
`drvblock.img`	Driver disk for block (storage) devices.
`drvnet.img`	Driver disk for network adapters.
`pcmciadd.img`	Driver disk for PCMCIA hardware.
`boot.iso`	All-in-one boot disk with drivers. While this doesn't fit on a single 1.44MB floppy, it can be installed on a mini-CD.

Note *This section is closely related to (and is somewhat repetitive of) the boot disk section in Chapter 4.If you're working with a computer with a PPC CPU, you can set your system to boot from the CD using the System Management Services menu. If you're working with an Itanium system, you can set this up through the EFI shell. If you're working with an IBM S/390, you can configure your system to boot from a Virtual Machine (VM) or a Logical Partition (LPAR), which is different from a standard PC hard disk logical partition. While this book focuses on installation on an x86 or AMD64 bit system, you can find more about other architectures through the Red Hat installation documents, available online at* `www.redhat.com/docs/manuals/enterprise`.

Creating a Boot or Driver Disk

Red Hat Enterprise Linux provides four utilities that help you create boot and driver floppies. Two of them (`dd` and `cat`) work in Linux; the other two (`RAWRITE.EXE` and `RAWWRITEWIN.EXE`) work in Microsoft Windows. The Linux utilities are standard commands you can run from other Linux or Unix computers; the image files and Microsoft utilities are available on the first Red Hat Enterprise Linux installation CD.

If you're currently running a Linux computer, use the following steps to create a boot disk. Remember, you'll probably also need one or more driver disks, as described in the following sections.

1. At a command-line interface, find the image files. For example, if you use the command

```
# mount /mnt/cdrom
```

   to mount the first Red Hat Enterprise Linux installation CD, the image files will be located in the `/mnt/cdrom/images` directory.

2. Insert a 1.44MB disk into a floppy drive. You don't need to use the `mount` command on that drive.

3. Use one of the following commands to convert the boot disk image, `bootdisk.img`, to a series of files on your floppy disk (`/dev/fd0` is the device associated with the first floppy drive on your computer):

```
# dd if=/mnt/cdrom/images/bootdisk.img of=/dev/fd0
# cat /mnt/cdrom/images/bootdisk.img > /dev/fd0
```

4. Repeat these steps with any driver disks you may require from the `images` directory.

If you're in Microsoft Windows and want to create a boot disk from the command-line interface, use the following steps to create that disk. Remember, you'll probably also need to repeat the process for one or both driver disks, as described in the following sections.

1. Insert the first Red Hat installation CD into a drive. These steps assume it's the F: drive, but if your drive letter is different, substitute accordingly.

2. Access a MS-DOS prompt. Select Start ➢ Run. In the Run dialog box that appears, type **`CMD`** in the text box and press Enter. This should open a command prompt window.

3. In the command prompt window, type **`F:`** and press Enter.

4. Start the `RAWRITE.EXE` utility, and run the following commands; insert a 1.44MB disk into your floppy drive when prompted:

```
F:\>/DOSUTILS/RAWRITE.EXE
Enter disk image source file name: /IMAGES/BOOTDISK.IMG
Enter target diskette drive: A:
Please insert a formatted diskette into drive A: and press -ENTER-:
```

5. Repeat the process with other required disk images in the `IMAGES` directory.

You can also use Microsoft Windows to create a boot disk by using the graphical `RAWWRITEWIN.EXE` utility. Remember, you'll probably also need one or both driver disks, as described in the sections that follow.

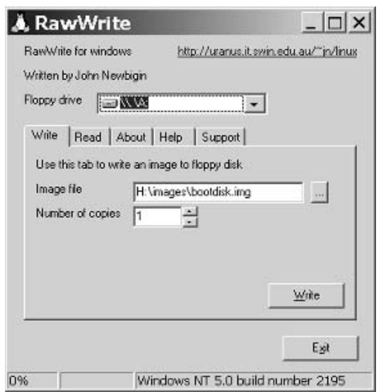
1. Insert the first Red Hat installation CD into a drive. These steps assume the CD is using the H: drive, but if your drive letter is different, substitute accordingly.

2. Access the utility. Open Microsoft Windows Explorer. Select Start ➤ Run. In the Run dialog box that appears, type **EXPLORER** in the text box and press Enter. This should open Microsoft Windows Explorer.

3. Navigate to the H: drive, and then access the `RAWRITEWIN` folder, which is inside the `DOSUTILS` folder. You can then double-click the `RAWWRITEWIN.EXE` utility. (Yes, the spelling of the `RAWRITEWIN` folder differs from the `RAWWRITEWIN.EXE` utility.)

4. This opens the RawWrite dialog box, shown in Figure 3.1. Click the Write tab if necessary. Click the button to the right of the Image File text box; you should be able to access the image file of your choice from the `H:\IMAGES` directory in the Open dialog box.

FIGURE 3.1
Creating a boot floppy with RawWrite



5. Insert a 1.44MB disk into the floppy drive, and click Write. Repeat this process with other required disk images.

Analyzing the Red Hat Boot Floppy

Whatever method you use to create it, the purpose and contents of the Red Hat Enterprise Linux 3 boot floppy remain the same. It's created from the `bootdisk.img` file in the `images` directory on the first Red Hat installation CD and is used to boot your computer. On this floppy, the `syslinux.cfg` file provides a roadmap to what comes next, as shown in Figure 3.2.

Take a careful look at this file. Table 3.2 describes the key commands and should help you interpret the `syslinux.cfg` file.

As you can see, the default is to load the compressed Linux kernel, `vmlinuz`, with the Initial RAM disk. As you can see in Figure 3.2, other options add different parameters.

TIP *You can use a Red Hat Enterprise Linux boot floppy or CD as a rescue disk. Using the techniques described in Chapter 11, it can help you recover from a number of failures, such as corrupted boot configuration files.*

FIGURE 3.2
The Linux boot roadmap

```
default linux
prompt 1
timeout 600
display boot.msg
F1 boot.msg
F2 options.msg
F3 general.msg
F4 param.msg
F5 rescue.msg
F7 snake.msg
label linux
  kernel vmlinuz
  append initrd=initrd.img
label text
  kernel vmlinuz
  append initrd=initrd.img text
label expert
  kernel vmlinuz
  append expert initrd=initrd.img
label ks
  kernel vmlinuz
  append ks initrd=initrd.img
label lowres
  kernel vmlinuz
  append initrd=initrd.img lowres
_
```

TABLE 3.2: COMMANDS IN *SYSLINUX.CFG*

COMMAND	DESCRIPTION
`default`	Specifies the default boot option, in this case, `default linux`.
`prompt`	Sets out the `boot:` prompt.
`timeout`	Configures the delay time, in tenths of a second, before the boot disk automatically starts the default option; normally set to 600, or one minute.
`display`	Points to the initial message file to display on the screen.
`Fx` *`option`*`.msg`	Sets the function key associated with a particular message file.
`label` *`command`*	Specifies the actions associated with a particular command.
`kernel`	Sets the name of the compressed kernel image on the boot disk.
`append`	Adds the parameters with which the boot disk loads the kernel.
`initrd`	Specifies the Initial RAM disk.
`text`	Starts installation in text mode; see Chapter 4.
`expert`	Starts installation in expert mode, where you specify the hardware drivers.
`ks`	Starts the installation with a Kickstart file; see Chapter 5.
`lowres`	Starts the installation in a low-resolution 640 × 400 graphics mode with a basic VESA (SVGA) driver.

Analyzing the Storage Device Driver Disk

You can't install Red Hat Enterprise Linux unless Anaconda detects a hard drive attached to your computer. The standard boot disk (from `bootdisk.img`) often recognizes standard IDE, SCSI, and even some USB hard drives that are connected to a PC; the required drivers are integrated into the compressed kernel.

That's why you may want to create a 1.44MB floppy from the `drvblock.img` file in the `images` directory of the first Red Hat Enterprise Linux installation CD. Use any of the techniques described earlier in this chapter to create this disk. It loads five files onto a floppy, which I briefly describe in Table 3.3.

TABLE 3.3: FILES IN THE STORAGE DEVICE DRIVER DISK (*DRVBLOCK.IMG*)

FILE	DESCRIPTION
`modinfo`	Contains a list of device drivers and descriptions
`modules.cgz`	Has a compressed version of all drivers listed in `modinfo`
`modules.dep`	Includes a list of dependencies, in other words, other drivers required by each device
`modules.pcimap`	Supports other PCI drivers and controllers
`pcitable`	Configures PCI settings for each device
`rhdd`	Labels this driver disk: "Supplemental Block Device Drivers"

Analyzing the Network Device Driver Disk

You can't install Red Hat Enterprise Linux over a network unless Anaconda detects a connected network card on your computer. The standard boot disk (from `bootdisk.img`) doesn't include any network drivers, so you need a supplemental driver disk for network installations.

You can create a 1.44MB floppy from the `drvnet.img` file in the `images` directory of the first Red Hat Enterprise Linux installation CD. Use any of the techniques described earlier in this chapter to create this disk. It loads six files onto a floppy, which are functionally similar to those on the storage device driver disk described in Table 3.3. While the contents have changed, the filenames are the same as previously.

Analyzing the PCMCIA Driver Disk

Installing Red Hat Enterprise Linux on a laptop computer often creates special issues. Laptops often rely on PCMCIA cards, as specified by the Personal Computer Memory Card International Association to connect to networks, SCSI devices, and more. These credit card–sized adapters are sometimes known as PC Cards. Naturally, Red Hat provides many of the major PCMCIA socket, network, and SCSI drivers in the `pcmciadd.img` file.

You can create a 1.44MB floppy from the `pcmciadd.img` file in the `images` directory of the first Red Hat Enterprise Linux installation CD. Use any of the techniques described earlier in this chapter to create this disk. It loads six files onto a floppy, which are functionally similar to those on the storage device driver disk described in Table 3.3. Although the contents have changed, the filenames are the same as previously.

The Boot ISO

One more file of note exists in the `/images` directory of the first Red Hat Enterprise Linux installation CD: `boot.iso`. You can create a boot CD from this 3MB file, using the techniques described for the `cdrecord` command in Chapter 14. It's suitable for simultaneous network installations where you don't want to run around loading and unloading boot and driver disks.

When you burn the `boot.iso` image onto a CD, the contents appear quite similar to the standard Red Hat boot floppy. It has two major differences: the files are all in an `isolinux` subdirectory, and the drivers associated with the aforementioned driver disks are combined in the Initial RAM disk image file, `initrd.img`.

Checking the Installation CDs

Before you start installing Red Hat Enterprise Linux from the installation CDs, you should check the integrity of those CDs to ensure that all the packages on the CDs are whole. One bad package out of the approximately 1,100 available on the Red Hat CDs can stop your installation cold.

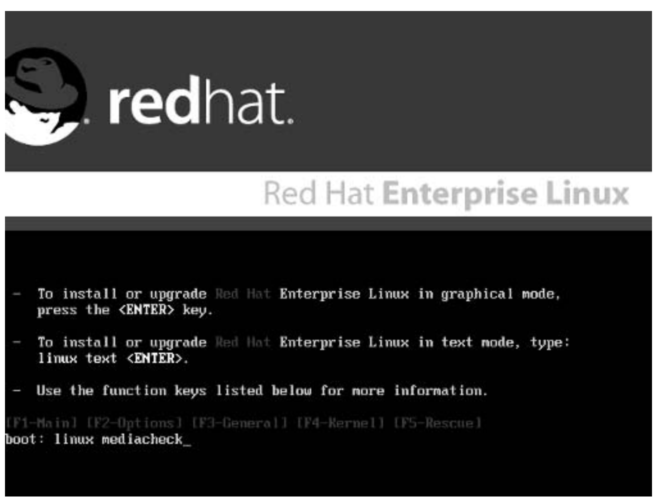
There are two basic options for checking your CDs. One involves starting the boot process with the `linux mediacheck` command; the other uses a statistical check based on the binary code on the CD.

NOTE *I've gone through the Red Hat installation process without the check—and after doing all the work required to configure Red Hat Enterprise Linux, I've seen an installation stop cold at the third CD because of a single bad package. I had no choice but to start from scratch.*

Inspecting CDs with *mediacheck*

To check your Red Hat Enterprise Linux installation CDs, boot your computer from a boot floppy or the first installation CD. Run the `linux mediacheck` command at the `boot:` prompt, shown in Figure 3.3.

FIGURE 3.3
Starting with a mediacheck



Anaconda then proceeds to install a generic kernel solely for the installation process. The first prompt that you see, shown in Figure 3.4, allows you to test the integrity of your CDs. If you select Skip, Anaconda proceeds to the installation process. Select OK; it's important to check your CDs.

NOTE *If you're using a CD created from a downloaded* `.iso` *file, you don't need to run the* `linux mediacheck` *command. You'll get the prompt shown in Figure 3.4 automatically. See the introduction for more information on downloading and creating Red Hat Enterprise Linux installation CDs.*

This mode isn't limited to the CDs for Red Hat Enterprise Linux 3. I used Red Hat Enterprise Linux 3's `mediacheck` feature to inspect a Red Hat Linux 9 installation CD. In the next screen (Figure 3.5), you can select whether to test the CD currently in the drive or eject it in favor of testing a different installation CD.

FIGURE 3.4
Anaconda offers to check your CD.

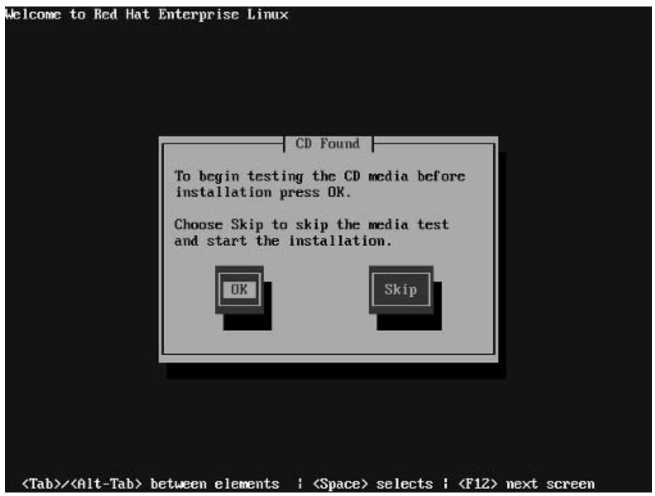


FIGURE 3.5
Ready to test

Insert the desired Red Hat installation CD, highlight Test, and press Enter. The test normally takes several minutes, at which point Anaconda identifies the CD and assigns it a grade of Pass or Fail. If necessary, reflect on the results and press Enter. Anaconda takes you to a slightly different screen, shown in Figure 3.6, where you can set up a different Red Hat installation CD for testing or insert the first Red Hat installation CD and then proceed with installation.

FIGURE 3.6
Continue testing or start the installation

Checking CDs with *md5sum*

You can directly check the MD5 signature associated with each Red Hat installation CD. MD5 is an algorithm for checking digital signatures. You can apply the `md5sum` command from within Linux to a downloaded Red Hat Enterprise Linux installation CD in `.iso` format. For example, if you've downloaded the first installation CD as a file named *`disc1`*`.iso`, run the following command:

```
# md5sum disc1.iso
abb3dd2cd1cd1b92b5e85b0d556b8e12 disc1.iso
```

The 32-digit alphanumeric number you get should match the number of the specified CD on your Red Hat download page. Some of the "rebuilds" include it in the `MD5SUMS-ftp.i386` file.

Installing Red Hat Enterprise Linux, Step by Step

Now that you've checked your CDs, you're ready to start installing Red Hat Enterprise Linux 3 on your computer. The actual installation process doesn't have to be nearly as complex as I portray in this chapter—I'm just trying to give you a feel for everything that Anaconda can do for you. In Chapter 5, I show you how to automate this process so you can install Red Hat Enterprise Linux on a number of computers simultaneously. But before you automate, you need to understand the details. I've divided the installation process into several sections, most of which I've described here.

- "Selecting Installation Prompt Options" describes what you can do at the first installation `boot:` prompt.
- "Configuring Basic Parameters" allows you to examine your choices with your keyboard and mouse, as well as the language used by Anaconda during the installation process.
- "Setting Up Hard Drives" shows in detail how you can configure different types of partitions in different formats using Disk Druid.
- "Configuring Installation Details" permits you to examine the nitty-gritty configuration details of the Red Hat Enterprise Linux installation.
- "Selecting Package Groups" takes a look at the various package groups that you can install with Red Hat Enterprise Linux as well as the individual package options.
- "Managing Post-Installation Steps" helps you configure the X Window and create a custom boot disk for your new system.

These steps assume you're installing Red Hat Enterprise Linux from the installation CDs. If you'd rather install Red Hat Enterprise Linux over a network connection, read Chapter 4.

These steps also assume you've already changed the initial boot sequence per Chapter 2 to boot from the first Red Hat installation CD. On an x86 or AMD64-based system, that's done through the BIOS. If you haven't, make sure this system at least boots first from your floppy drive, or create a boot disk.

Red Hat Enterprise Linux 3 Workstation

The software included with Red Hat Enterprise Linux 3 WS (Workstation) is nearly identical to what you'll find in a Red Hat Enterprise Linux 3 server. In fact, the only difference is in the first CD. The server version includes 22 server packages that you won't find on the workstation. They include the following:

- AMANDA, the Advanced Maryland Automatic Network Disk Archiver
- DNS (Domain Name Service) server
- DHCP server
- FreeRADIUS, which is commonly used by Internet Service Providers
- News message services
- Kerberos 5 server
- Lightweight Directory Assistance Protocol (LDAP) services
- Diskless workstation server
- Tux Web server
- Very Secure FTP service
- Network Information Service (NIS) Server

As you can see, this list doesn't include all services associated with Linux. For example, you can still configure services such as web services, Samba, NFS, and more withRed Hat Enterprise Linux 3 Workstation.

NOTE *It's possible to start the graphical installation process over an NFS (Network File System) connection. For more information on setting up an NFS-based network installation, see Chapter 4, and then return here for the graphical installation steps.*

Selecting Installation Prompt Options

You can start the installation process by booting your computer from one of two sources: the first Red Hat installation CD or a boot disk. Either media will get you to the same start screen, shown in Figure 3.7. When you see this screen, press F2 within 60 seconds. Otherwise, Red Hat Enterprise Linux starts graphical-mode installation automatically.

FIGURE 3.7
The installation prompt

As you can see, several menus are available. We'll examine the different *installation* screens. The selections on the first screen are basic: you can choose to install Red Hat Enterprise Linux in graphical or text mode. We focus on graphical mode in this chapter and text mode in Chapter 4. When you press F5, Anaconda takes you to the Rescue Mode Help screen, which is unrelated to installation and is covered in Chapter 11.

When you're ready, you can press Enter to start graphical-mode installation, or type **`text`** and press Enter to start text-mode installation. I think installation over a network is more efficient, especially in the enterprise. I show you how to set this up in Chapter 4.

However, if you have problems during installation, you may want to start again and try something else. Therefore, proceed to the "Installer Boot Options" section to examine the variety of commands you can run at the `boot:` prompt.

Installer Boot Options

There are a number of different ways to install Red Hat Enterprise Linux. You can disable probing of troublesome hardware, and you can set up a network installation from the first CD. To see some

of these options, press F2. This takes you to the Installer Boot Options menu shown in Figure 3.8. Different options from this menu are briefly described in Table 3.4.

FIGURE 3.8
Installer Boot Options menu

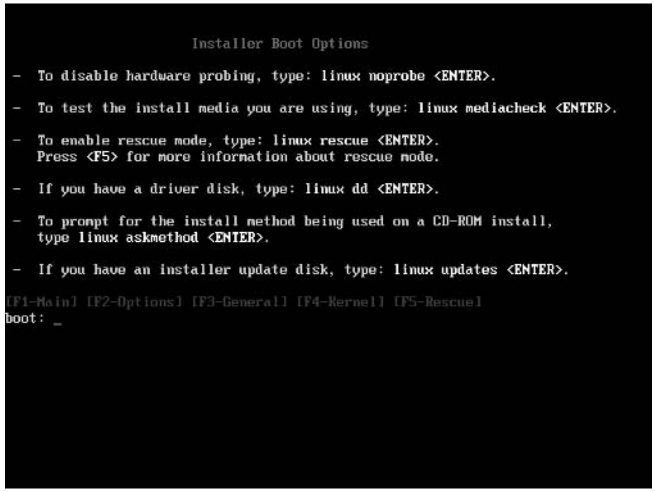


TABLE 3.4: INSTALLER BOOT OPTIONS

Option	**Description**
`linux noprobe`	Starts the installation process without automatic hardware detection; you'll need to select the drivers for any SCSI hard disks and network cards from a list.
`linux mediacheck`	Begins the installation process with the text-mode prompts that allow you to check the integrity of the Red Hat installation CDs; by default, continues in graphical mode.
`linux rescue`	Boots a basic Linux system in rescue mode that tries to detect a current Linux installation. See Chapter 11 for more information.
`linux dd`	Starts the installation process with a prompt for a driver disk; useful for third-party drivers.
`linux askmethod`	Begins the installation process; allows you to select the language and keyboard and then allows you to select from local or network installation options.
`linux updates`	Supports an update of current packages using a custom installer update disk.
`linux lowres`	Starts the installation in low-resolution graphics mode, 640 × 400, also known as VGA (Video Graphics Adapter).

You can run any of these installation options in text mode; just substitute **`text`** for **`linux`**.

NOTE *Previous versions of Red Hat Linux allowed you to configure the ReiserFS filesystem during installation if you started the process with the* `linux reiserfs` *command. ReiserFS commands are part of the kernel-unsupported RPM and can be configured only after Red Hat Enterprise Linux 3 is installed.*

General Boot/Kernel Parameter Help

Some hardware requires a little help during the installation process. The General Boot Help menu can help you define some parameters to use for your graphics hardware. Press F3 in the Red Hat Enterprise Linux installation start screen to access this menu, shown in Figure 3.9.

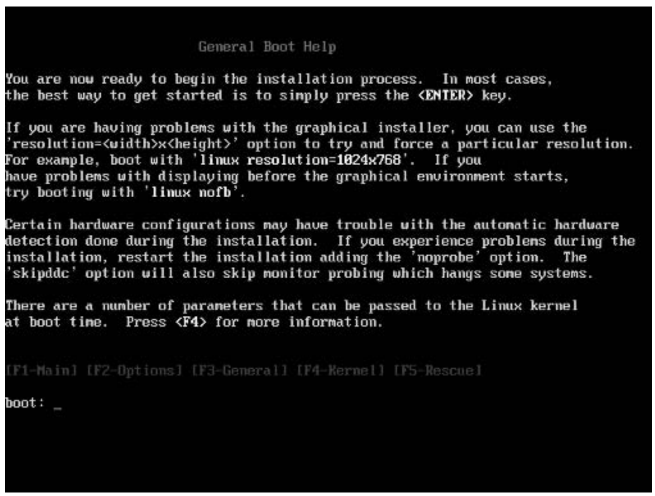


FIGURE 3.9
General Boot Help menu

There are additional arguments that you can add after `linux` or `text` at the `boot` prompt. For example, the `linux upgradeany` command searches for and offers to upgrade any computer where Linux is detected, independent of what may be found in the `/etc/redhat-release` file. There is a lot more that you can do at the `boot:` prompt, as described in Table 3.5.

TABLE 3.5: ***BOOT:*** **PROMPT INSTALLATION ARGUMENTS**

ARGUMENT	DESCRIPTION
`apm=off`	Disables Advanced Power Management (APM) during the installation process.
`display=`*`ip_addr`*`:0`	Forwards the installation display to a computer with an IP address of *`ip_addr`*. To make this work, be sure the receiving computer allows remote X Window access; see Chapter 15 on the `xhost` command for more information.

TABLE 3.5: *BOOT:* PROMPT INSTALLATION ARGUMENTS *(continued)*

Argument	Description
`expert`	Prompts for a driver disk; supports partitioning of removable drives. If you're installing on a SCSI hard drive, you'll need to supply at least the associated driver disk.
`ide=nodma`	Disables DMA addressing on IDE devices such as hard drives.
`isa`	Prompts you to confirm that Anaconda has detected the correct ISA drives or similar devices.
`mem=`*`xyz`*`M`	Assigns a specific amount of RAM.
`nmi_watchdog=1`	Adds kernel-debugging messages in one of the message screens described later.
`nopcmcia`	Avoids installing PCMCIA controllers; if you're installing Red Hat Enterprise Linux from a CD that's not controlled through a PCMCIA connection, you don't need Anaconda to look for the PCMCIA controller.
`nousb`	Keeps Anaconda from installing USB support.
`reboot=b`	Modifies the kernel reboot method; some installations may otherwise hang just before the final step.
`resolution=`*`axb`*	Specifies an installation video mode such as 640 × 480 or 1024 × 768.
`serial`	Starts serial console support during installation.
`skipddc`	Avoids the `ddcprobe` command, which is otherwise used to detect the monitor and graphics card. See Chapter 29 for more information on `ddcprobe`.
`upgradeany`	Looks for Linux installations to upgrade, independent of the contents of `/etc/redhat-release`.

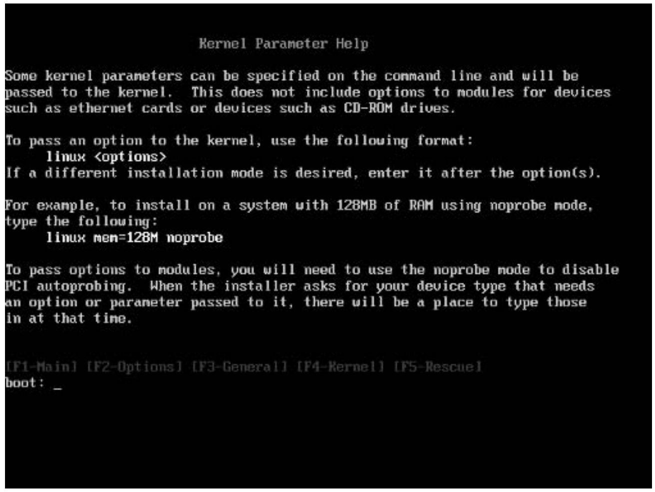
NOTE *The* `apic` *argument that supported installation on computers with the Intel 440GX chipset BIOS is no longer available or required; support is now set up automatically.*

There are other options described in the Kernel Parameter Help menu, shown in Figure 3.10. Press F4, to review this menu. It includes information similar to the General Boot Help menu; Tables 3.4 and 3.5 include some related arguments that you can pass to the kernel. Some are direct, such as `mem=256M`; others, such as `noprobe`, work indirectly by allowing you to specify the hardware address of key components of your PC.

Configuring Basic Parameters

Now we're actually ready to start the installation. Unless you have specific issues addressed by the previous section, just press Enter at the installation `boot:` prompt to start the Red Hat Enterprise Linux installation process in graphical mode.

FIGURE 3.10
Kernel Parameter Help menu



NOTE *If you've started the installation by entering* **`linux mediacheck`** *at the* `boot:` *prompt, or you're starting from a CD created from a downloaded* `.iso` *file, Anaconda prompts you to check the integrity of your CDs, as described earlier.*

Anaconda probes your system to see if it meets the requirements for a graphical installation. This shouldn't be a problem if you have the minimum supported RAM on your system, 256MB. (Graphical installations are actually possible with lesser amounts of RAM).

TIP *If you're configuring a computer with 256MB (or less) of RAM, you may get a message during the boot process warning you of this problem. Some memory may be shared with the video system; small amounts of memory may be taken by a dual-boot configuration.*

Next, it checks your system for the other requirements associated with a graphical installation: a video card, monitor, and mouse. You should see messages similar to the following:

```
Running anaconda, the Red Hat Enterprise Linux system installer - please wait...
Probing for video card: Intel 810
Probing for monitor type: S/M 955DF
Probing for mouse type: Generic - Wheel Mouse (PS/2)
Attempting to start native X Server
Waiting for X server to start...log located in /tmp/X.log
1...2...3...4...5.... X server started successfully.
```

The messages you see list the hardware detected by Anaconda. If you have problems, note the location of the log file: `/tmp/X.log`. This message is a little unusual; the file actually disappears once Red Hat Enterprise Linux is installed. We'll take a look at this file shortly.

If the hardware on your system passes the test, you'll see the first Anaconda installation screen, shown in Figure 3.11.

FIGURE 3.11
The graphical installation begins.

The basic graphical installation screen includes some help notes in the left pane. If you click Release Notes, this opens the Release Notes window, shown in Figure 3.12. This includes the test from the `RELEASE-NOTES` file on the first Red Hat Enterprise Linux installation CD.

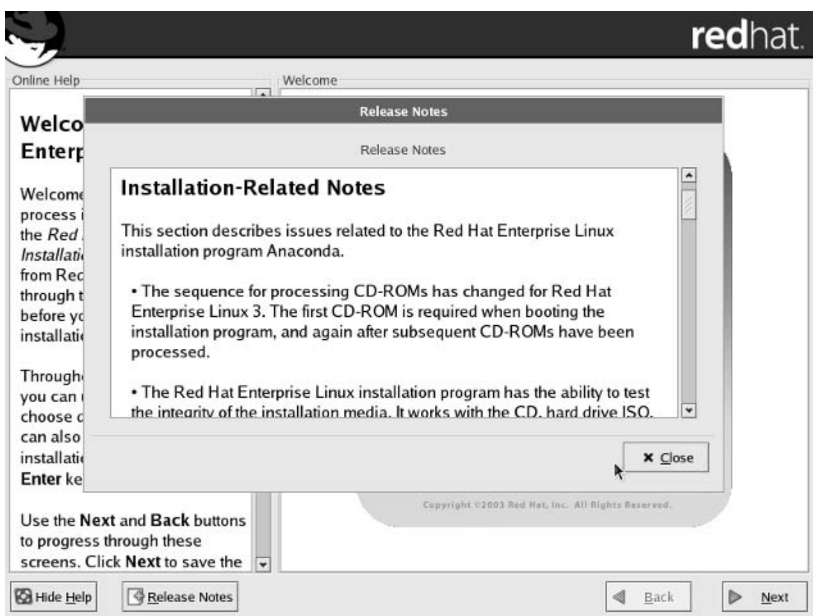
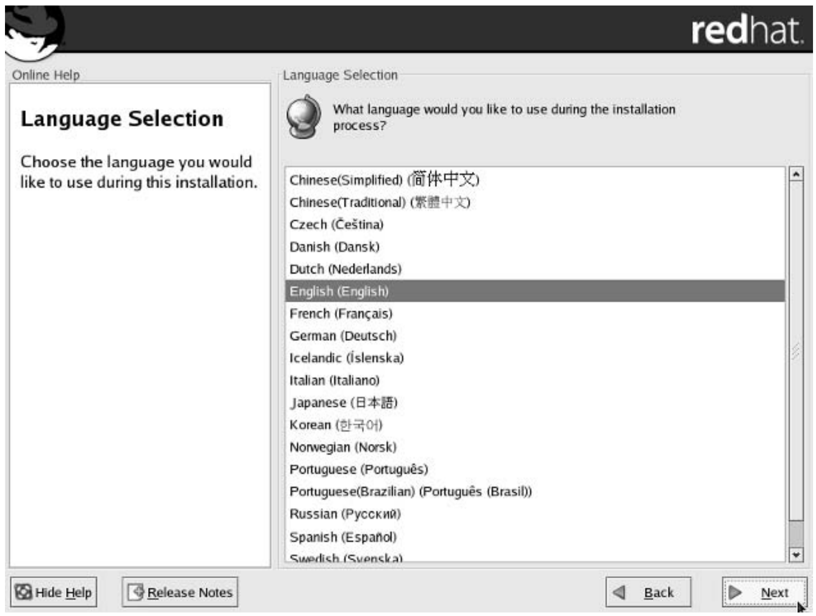


FIGURE 3.12
Release Notes window

Read the Release Notes. They can help you learn more about Red Hat Enterprise Linux. When you've finished, click Close to exit the release notes, and then click Next to continue. Anaconda takes you to the Language Selection screen, shown in Figure 3.13, which lets you select from 19 languages or dialects for the remainder of the installation process. This does not determine the languages that are loaded or used once Red Hat Enterprise Linux is installed; we'll look at that step later. The rest of this chapter assumes that you're proceeding in English. Click Next to continue.

FIGURE 3.13
Selecting an installation language



Now we'll look at the Keyboard Configuration screen shown in Figure 3.14, which lets you select from 55 types of keyboards for your system. If Anaconda detected your keyboard, it should be highlighted. Your selection determines the default keyboard once Red Hat Enterprise Linux is installed. You can change the default keyboard after installation by using the `redhat-config-keyboard` utility described in Chapter 2. Select the keyboard that most closely matches your system and click Next to continue.

Next, examine the Mouse Configuration screen, shown in Figure 3.15. This screen title is misleading; you can configure several different types of *pointing devices* with Anaconda.

NOTE *If you've set up a graphical network installation, you'll first select an installation language and keyboard in text mode, as described in Chapter 4. Once your computer is connected to the network installation source, it allows you to select a pointing device.*

FIGURE 3.14
Choosing a keyboard

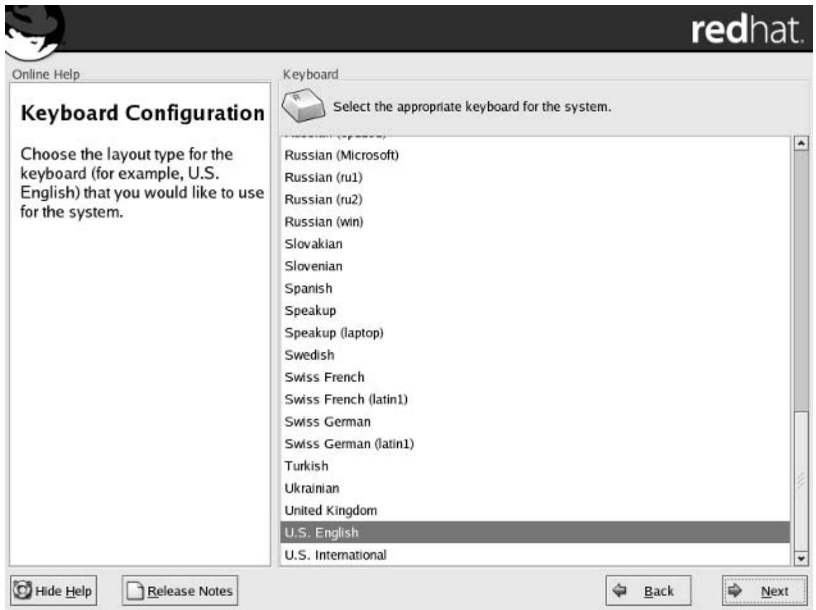
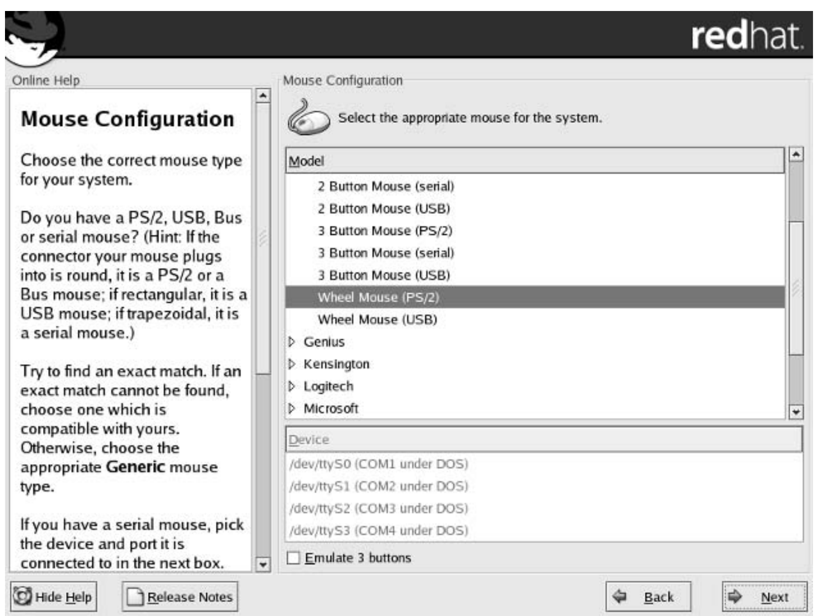


FIGURE 3.15
Selecting a pointing device



A pointing device can be a mouse, a touchpad, a trackball, or even a tablet. Red Hat Enterprise Linux can even work with pointing devices connected through a USB (Universal Serial Bus) port. If you're configuring a pointing device that is connected to a serial port, the Device text box is active, and you can select the appropriate serial port device.

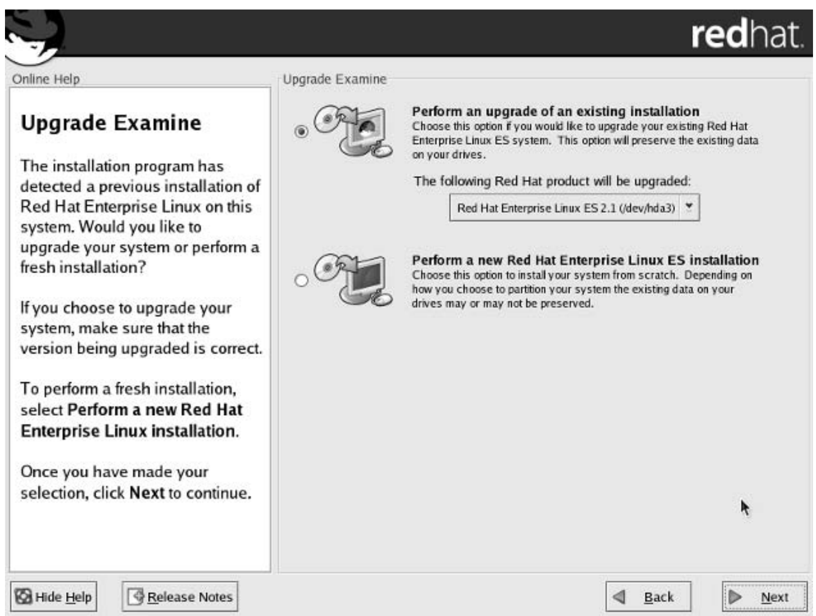
If you have a two-button mouse, you should activate the Emulate 3 Buttons option. This allows you to simulate the functionality of a middle mouse button by pressing both buttons together. However, if you have a mouse wheel, try pressing it. If it clicks, Red Hat may already recognize it as a middle button.

If Anaconda detected your pointing device, it should be highlighted on your screen. You can change the default pointing device after Red Hat Enterprise Linux is installed with the `redhat-config-mouse` utility described in Chapter 2. Select the pointing device that most closely matches your system and click Next to continue.

If you're installing on a computer that includes a previous version of Red Hat Enterprise Linux, you may see an Upgrade Examine screen. Upgrades are covered near the end of this chapter. If you see the screen shown in Figure 3.16, select Perform A New Red Hat Enterprise Linux Installation and click Next to continue.

NOTE *Officially, while Red Hat supports upgrades on x86 systems, it recommends a fresh installation, even over an existing installation of Red Hat Enterprise Linux 2.1.*

FIGURE 3.16
Installing, not upgrading



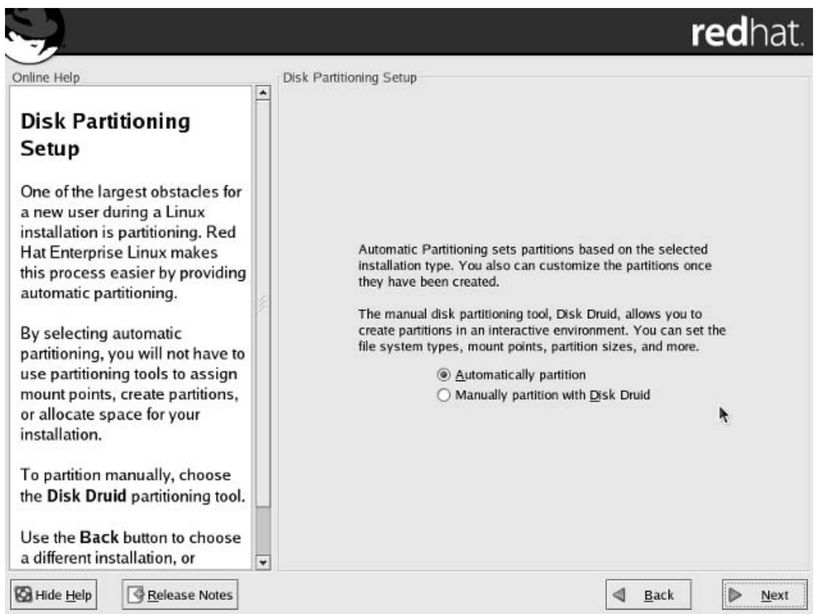
If you're experienced with Red Hat Linux, you may be expecting to select an installation type at this point. For example, Red Hat Linux 9 allows you to select a Personal Desktop, Workstation, or Server installation. For Red Hat Enterprise Linux 3, that's predetermined by whether you've started from a WS (workstation), ES (entry-level server), or AS (advanced server) installation CD.

NOTE *If you're using one of the rebuilds, you'll still see a screen where you select from default installations such as a workstation or a server.*

Setting Up Hard Drives

In the next several steps, we'll set up partitions on selected hard drives connected to your computer and recognized by Linux. Once you've selected an installation type, you get to choose whether to let Anaconda set up partitions for you or to proceed directly to Disk Druid. This screen is shown in Figure 3.17.

FIGURE 3.17
Choosing automatic or manual partitions



We'll select automatic partitioning and then continue on to Disk Druid to illustrate what Anaconda can do for you. If you select Manually Partition With Disk Druid, Anaconda skips the next step. Make your selection and click Next to continue.

Anaconda asks for your input as to where it should apply automatic partitions. As you can see in Figure 3.18, you have several options, which are explained in Table 3.6.

WARNING *If you're installing Red Hat Enterprise Linux ES or AS and want to dual-boot with another operating system such as Microsoft Windows, pay attention! Anaconda defaults to the Remove All Partitions On This System option, which would delete all Microsoft Windows partitions on your computer.*

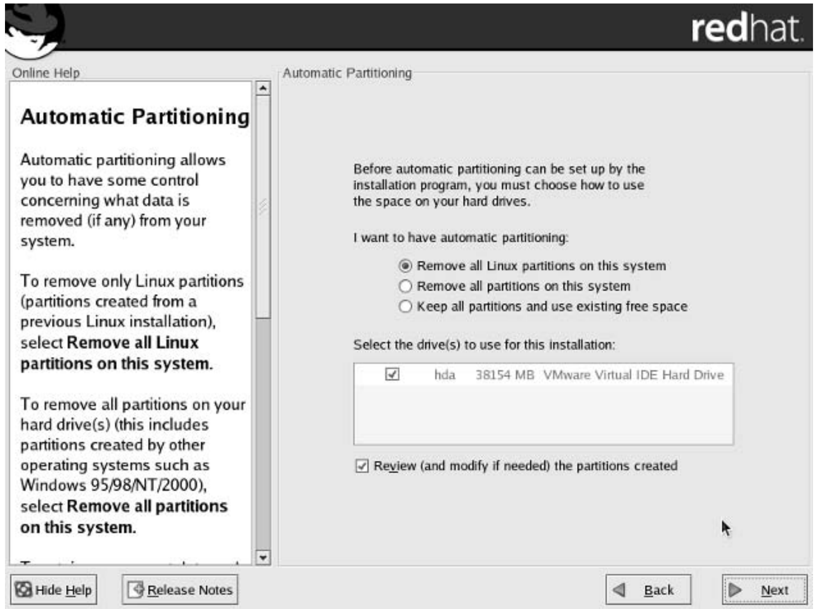


FIGURE 3.18
Configuring automatic partitioning

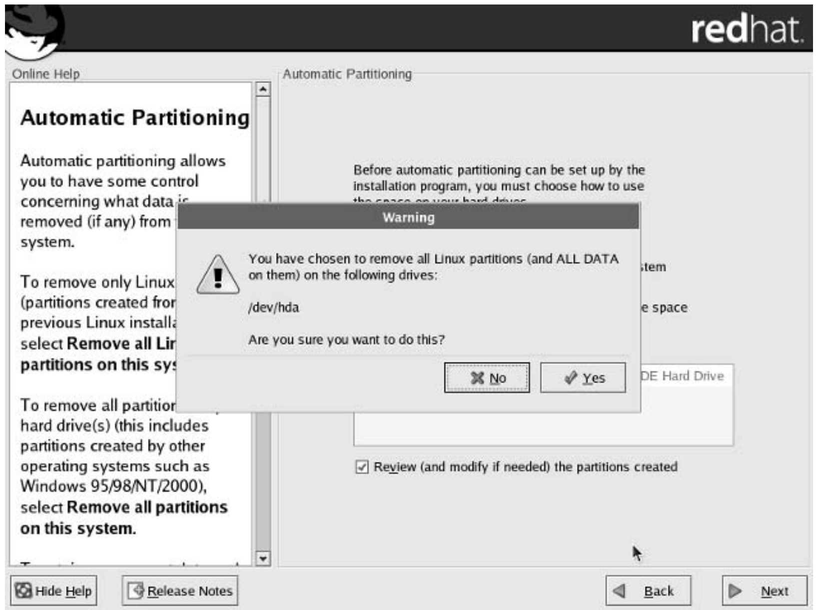
TABLE 3.6: AUTOMATIC PARTITIONING OPTIONS

OPTION	DESCRIPTION
Remove All Linux Partitions On This System	Deletes all currently configured partitions that are formatted to Linux filesystems. Applied to all selected hard drives.
Remove All Partitions On This System	Deletes *all* partitions on the selected hard drives. If you have another operating system such as Microsoft Windows, this action deletes that operating system.
Keep All Partitions And Use Existing Free Space	Doesn't delete any partitions. Attempts to configure partitions for Red Hat Enterprise Linux in any available unallocated hard drive space.
Select the Drive(s) To Use For This Installation	Lists the recognized hard drives on your computer. Automatic partitioning applies only to the drives that you select. Device names such as hda are explained in Chapter 2.
Review (And Modify If Needed) The Partitions Created	If checked, the next installation step illustrates Anaconda's proposed partition configuration in Disk Druid.

For the purpose of this installation, proceed by selecting Remove All Linux Partitions On This System. Also select the option Review (And Modify If Needed) The Partitions Created. (If you don't select this option, Anaconda skips the upcoming Disk Druid menu.) Make your selections, and click Next to continue.

Before Anaconda removes any partitions, it sends you a warning message. If you've directed Anaconda to delete Linux partitions, the message is shown in Figure 3.19.

FIGURE 3.19
You'll see this warning before Anaconda removes partitions.



If you've directed Anaconda to delete all partitions, or if the drive is new, you should get a similar warning. Make sure you're actually ready to delete the noted partitions. If you're ready, click Yes to continue. If you have partitions on more than one hard drive, you'll be asked to confirm again for the other drives.

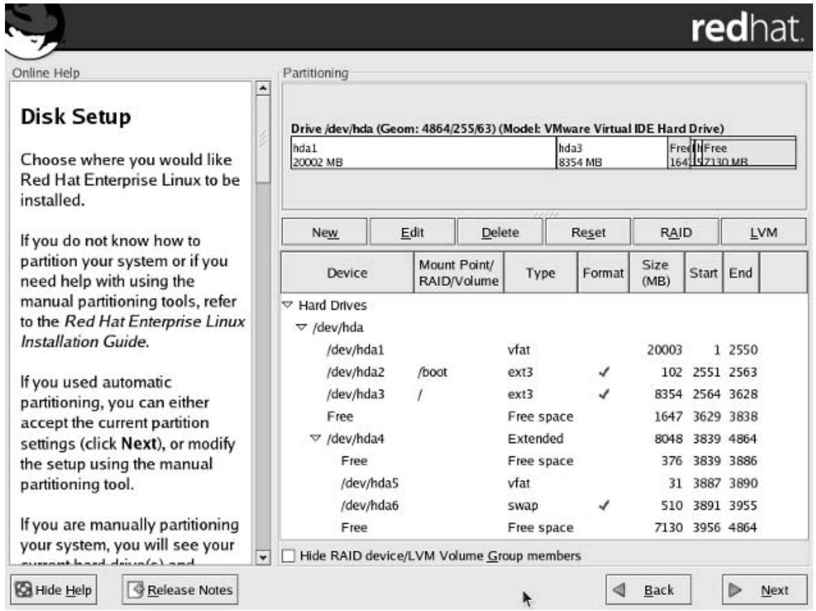
NOTE *You may also get a warning about a* `/boot` *partition. If you already have another operating system, such as Microsoft Windows, on your computer, Anaconda probably can't install the* `/boot` *partition in the most desirable area of your hard drive, below the 1,024th cylinder. The BIOS on some older computers won't be able to find your Linux boot files if the* `/boot` *partition is located above this cylinder on your hard drive. Even if your computer is affected, there are at least two ways to work around this issue. You can boot Linux from a boot floppy, or you can install a third-party bootloader such as Partition Magic or System Commander.*

NOTE *The boot partition you need to configure on an Itanium system is a little odd; it's set on the* `/boot/efi` *directory in a partition formatted to the Microsoft-style VFAT filesystem.*

Setting Up Partitions with Disk Druid

Disk Druid is Anaconda's semi-automated disk-partitioning utility. The results of Disk Druid's automatic partitioning on this desktop computer are shown in Figure 3.20.

FIGURE 3.20
Disk Druid at work



As you may guess, this is a dual-boot installation; `/dev/hda1` happens to be a partition formatted to Microsoft's FAT32 filesystem. The swap partition, configured on `/dev/hda6`, is twice the size of the RAM on this desktop computer.

NOTE *If you have Microsoft Windows on the computer where you're installing Red Hat Enterprise Linux, Anaconda shows both FAT16 and FAT32 partitions as type vfat.*

The Disk Druid screen is organized into sections. The top includes a map of current partitions as configured on recognized hard drives on your computer. It's followed by a series of command buttons that you'll explore momentarily. The bottom of the screen includes data on each drive and partition, as explained in Table 3.7.

TABLE 3.7: DISK DRUID DRIVE DEFINITIONS

COLUMN	DESCRIPTION
Device	Lists the device file for each hard drive and partition
Mount Point/RAID/Volume	Specifies the directory mounted on the partition
Type	Notes the filesystem of the partition
Format	Specifies drives to be formatted (if checked)
Size (MB)	Lists the size of the partition, in megabytes
Start	Notes the starting cylinder of the partition
End	Notes the ending cylinder of the partition

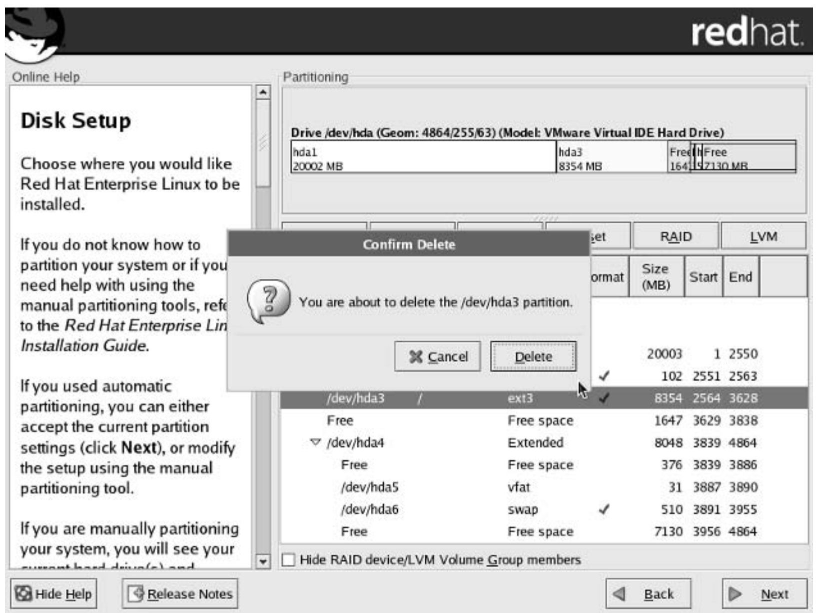
At the bottom of the screen, you can choose to Hide RAID Device/LVM Group Members. If you activate this option, the partitions in a RAID (Redundant Array of Independent—or Inexpensive—Disks) array or a Logical Volume Group aren't shown. For more information on RAID, see Chapter 14; for more information on LVM, see Chapter 7.

Now let's examine each of the command options shown in Figure 3.20. We'll use the configuration shown in the following sections to illustrate our discussion. Since there is currently no room on the hard drive shown, we'll start by deleting a partition.If you have a computer with a BIOS, you probably shouldn't configure a `/boot` partition on a RAID array. It's generally not supported.

Deleting a Partition

To delete a partition, highlight it and click Delete. In the example shown in Figure 3.20, we've highlighted `/dev/hda3` and clicked Delete. Before Disk Druid deletes the partition, it asks for confirmation, as shown in Figure 3.21.

FIGURE 3.21
Confirming a deleted partition



Now we have additional free space available from the deleted partition.

Adding a Partition

You need free space on the available hard drives before you can add a partition. If you have free space on your hard drive, click New. This opens the Add Partition dialog box, shown in Figure 3.22. Each item in the figure is explained in Table 3.9.

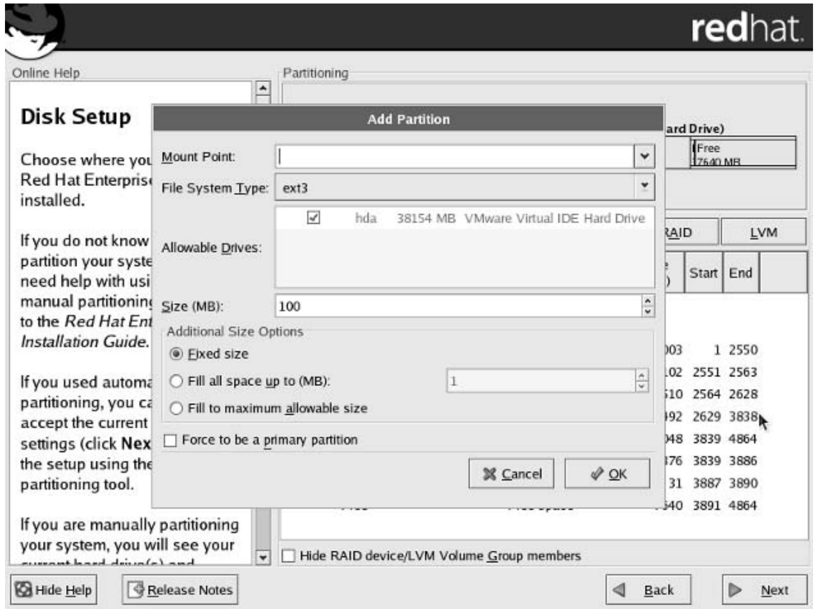


FIGURE 3.22
Adding a partition

TABLE 3.8: OPTIONS IN THE ADD PARTITION DIALOG BOX

OPTION	**DESCRIPTION**
Mount Point	Specifies the directory to be mounted on the partition; for mountable directories, consult the discussion on the Filesystem Hierarchy Standard in Chapter 7. This isn't applicable if the filesystem type is LVM, RAID, or swap.
File System Type	Sets the format for the partition; you're allowed to select form the Linux ext2 or ext3 standard, the Linux swap format, a LVM physical volume, a software RAID volume, or a Microsoft Windows–style VFAT format.
Allowable Drives	Notes the hard drive device associated with the partition.
Size (MB)	Specifies the size of the partition, in megabytes.
Fixed Size	Sets the partition size as specified.
Fill All Space Up To (MB)	If there's free space on your hard drive, the size of this partition grows up to the specified limit.
Fill To Maximum Allowable Size	Fills any remaining free space on the hard drive.
Force To Be A Primary Partition	Generally, you'll want the partition with the `/boot` directory to be on a primary partition below cylinder 1024.

TIP *When you set up partitions on a hard drive, remember the limit of 16 partitions. If you exceed this limit, you won't find the problem until after it looks as if installation is complete.*

For the purpose of this chapter, I've added four LVM partitions, four software RAID partitions, and a root (/) directory partition in the remaining space. You'll see how this works when we demonstrate what you can do when you click the RAID and LVM buttons.

Editing a Partition

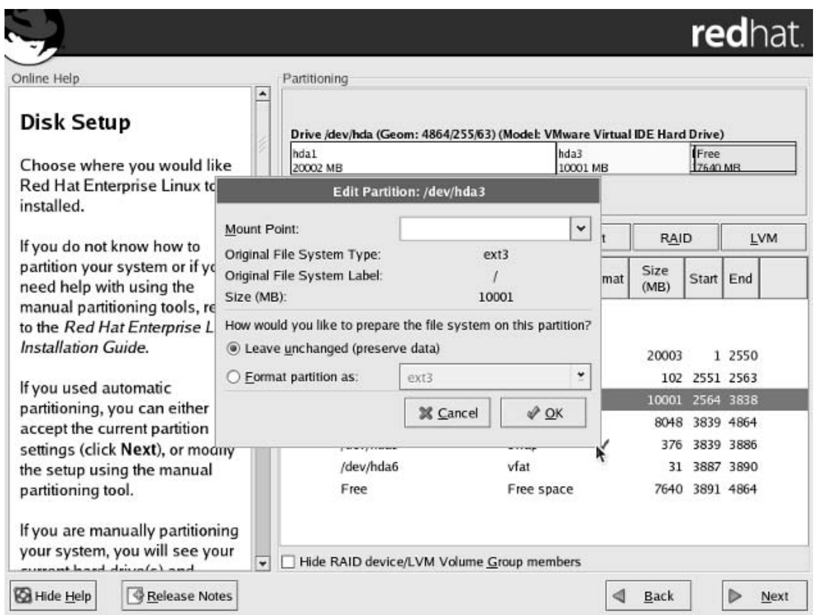
Editing a partition is similar to adding a partition. For example, Figure 3.23 illustrates what happens when I highlight and edit the partition with my root (/) directory. The screen contains information identical to that in Figure 3.22. Please refer to Table 3.8 for details on the Edit Partition window.

WARNING *Anaconda will not install Red Hat Enterprise Linux 3 on hard drives with bad blocks.*

Resetting the Partition Table

Any changes you make aren't written until the partitions are formatted. If you want to return to the original partition table on your hard disk, click Reset. You'll get a chance to confirm your intent. Once you do, the partition table reverts to the configuration when you started Disk Druid.

FIGURE 3.23
Editing an existing partition



Making RAID

Once you've configured software RAID partitions, you can create a RAID array. Ideally, the software partitions in a RAID array should be on different physical hard drives. Then the failure of one hard drive doesn't destroy your data in a RAID 1 or RAID 5 array. For more information, see Chapter 14.

Click RAID. Disk Druid takes you to the RAID Options dialog box, shown in Figure 3.24. As you can see, this window contains three options, which are described in Table 3.9.

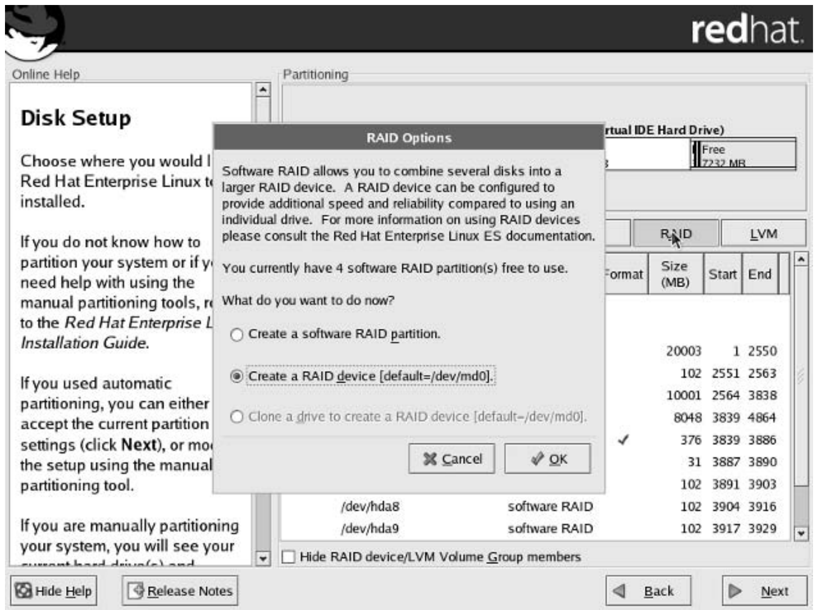


Figure 3.24
Disk Druid software RAID options

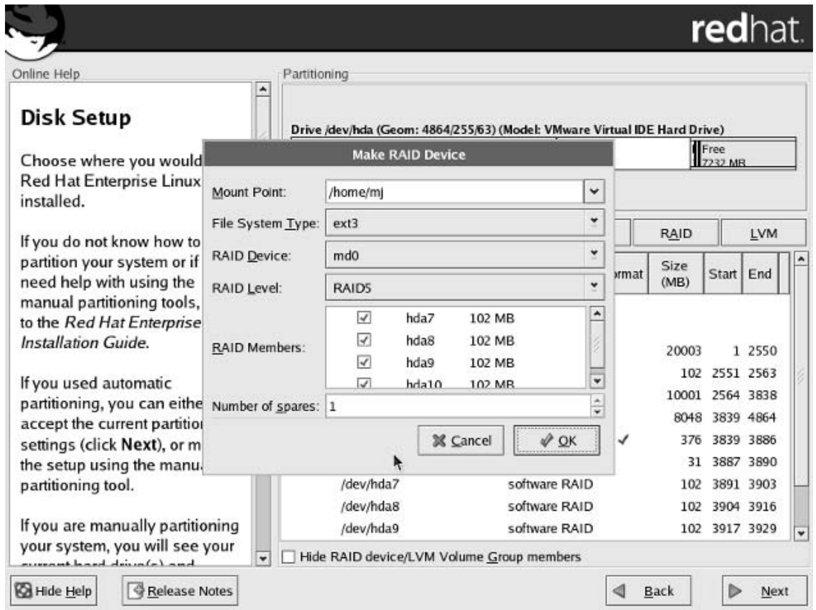
Table 3.9: Software RAID Configuration Menu Options

Option	**Description**
Create A Software RAID Partition	Opens the Add Partition window with a software RAID filesystem type.
Create A RAID Device	Opens the Make RAID Device window, where you can assign software RAID-formatted partitions to a RAID device.
Clone A Drive To Create A RAID Device	If you have two different physical hard drives, you can clone a RAID device from one drive to the other.

You already learned how to create a software RAID partition in the "Adding a Partition" section. If you have more than one hard drive on your computer and want to get serious about RAID, I recommend that you read one of the hardware RAID HOWTOs at `www.tldp.org`.

For the purpose of this installation, I've selected the Create A RAID Device option. After I click OK, Disk Druid takes me to the Make RAID Device dialog box, shown in Figure 3.25.

FIGURE 3.25
Making a RAID device



As shown in the figure, I've created a RAID device for the `/home/mj` directory. This is a RAID 5 device, formatted to the Linux ext3 filesystem. Since RAID 5 requires a minimum of three member partitions, it's possible to set this up with one spare partition. If one partition goes bad, RAID 5 will rebuild the required data on the spare partition automatically.

MAKING LVM

Once you've configured LVM physical volumes, you can create a LVM volume group. LVM is more practical on a single hard drive. As you can add and delete LVM physical volumes from a group, you can grow or compress the size of a partition assigned to a directory such as `/usr`. For more information, see Chapter 7.

Click LVM. Disk Druid takes you to the Make LVM Volume Group dialog box, shown in Figure 3.26. Each configurable option in the figure is explained in Table 3.11.

TABLE 3.10: MAKE LVM VOLUME GROUP OPTIONS

OPTION	DESCRIPTION
Volume Group Name	Sets the name of the LVM volume group
Physical Extent	Specifies the chunk of disk space associated with this volume group
Physical Volumes To Use	Lists LVM-formatted physical volumes (PVs)

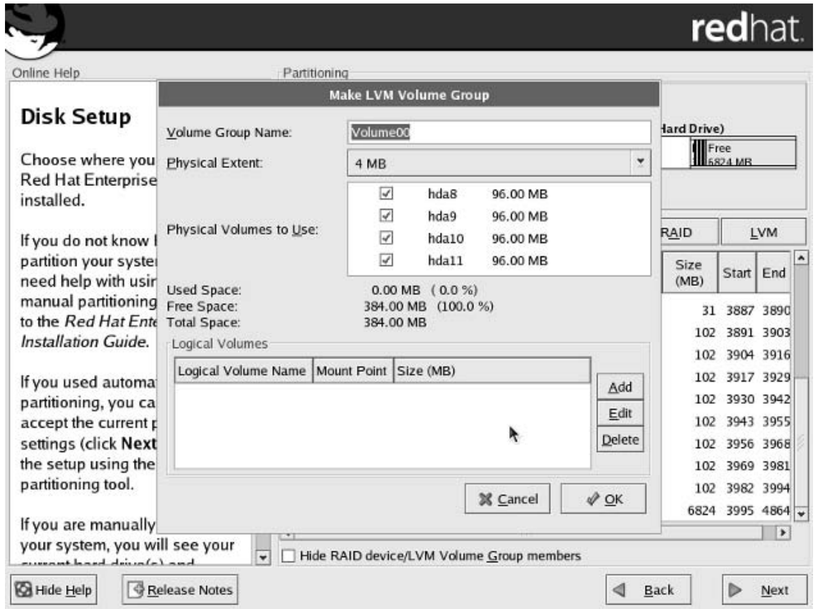


FIGURE 3.26
Making a LVM volume group

Make sure the physical volumes you want to use for this volume group (VG) are checked. Name the volume group, and then click Add. This opens the Make Logical Volume dialog box, shown in Figure 3.27; the options are described in Table 3.12.

TABLE 3.11: MAKING A LOGICAL VOLUME

OPTION	DESCRIPTION
Mount Point	The directory to be mounted on the logical volume (LV)
File System Type	The format associated with the LV
Logical Volume Name	An arbitrary name for the LV
Size (MB)	The size to be allocated to the LV, which includes the PVs that you've added to the LV

Once you've created the LV, click OK. This returns you to the Make LVM Volume Group window. You'll note that the amount of free space is reduced by the PVs you've allocated to the new LV. When you've finished creating LVs, click OK to return to the main Disk Druid window.

LEAVING DISK DRUID

The final result is shown in Disk Druid, which includes your new partitions, RAID device arrays, and LVM volume groups, as shown in Figure 3.28. When you're ready, click Next to move beyond Disk Druid.

FIGURE 3.27
Making a logical volume

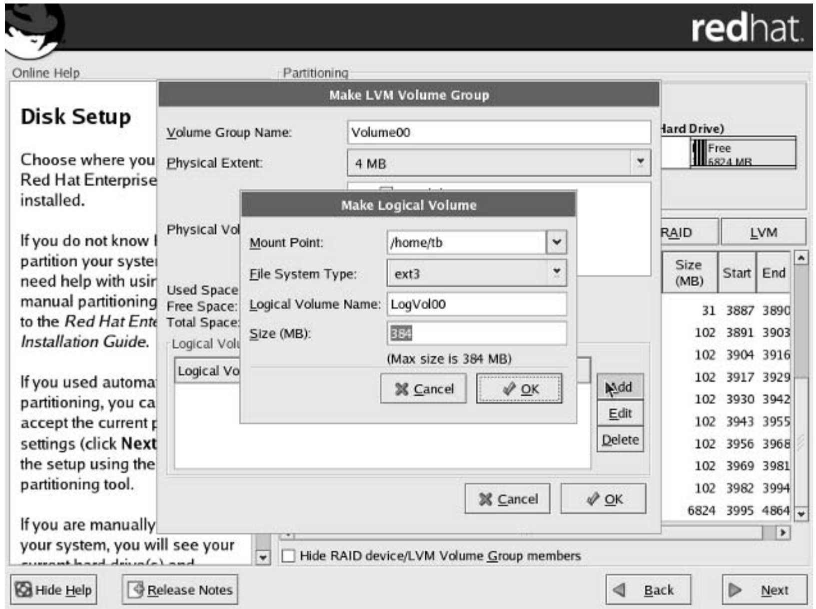
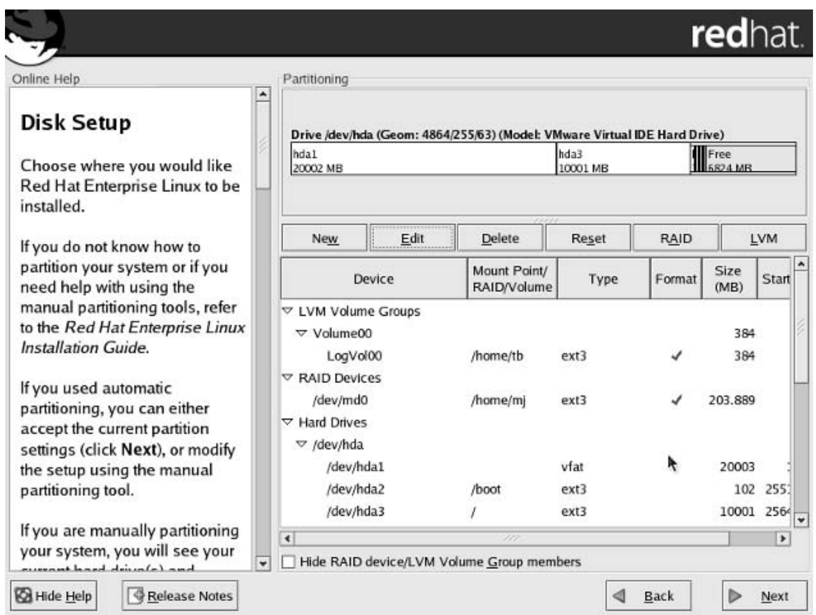


FIGURE 3.28
Disk Druid displays the revised partition table.



If you've selected preexisting partitions for your new installation, you'll get a warning that certain partitions are about to be formatted.

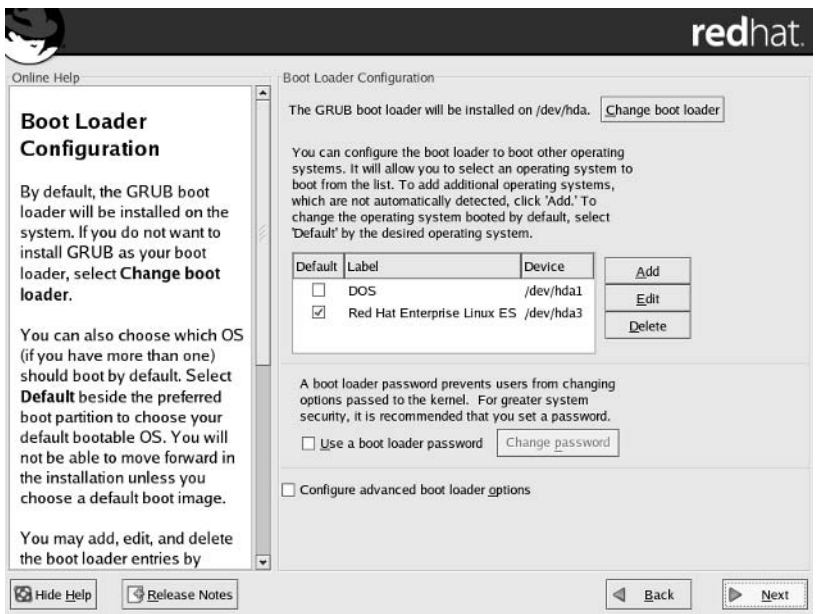
Configuring Installation Details

Now we'll explore the details of the Red Hat Enterprise Linux installation process between Disk Druid and package group selection. The topics are wide and varied, starting with bootloader configuration and ending with authentication configuration.

Bootloader Configuration

After you finish going through Disk Druid, it's time to configure the bootloader, which is what you see when you first boot your computer. Anaconda takes you to the Boot Loader Configuration screen, shown in Figure 3.29.

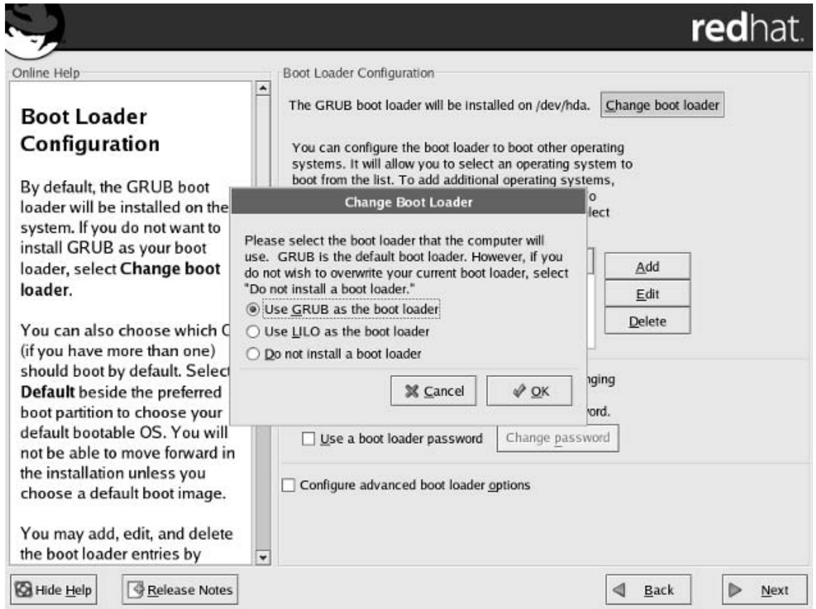
Figure 3.29
Boot Loader Configuration screen



The default is GRUB, the Grand Unified Bootloader. The message at the top of the screen tells you that GRUB will be installed on the Master Boot Record (MBR) of the first IDE hard drive (`/dev/hda`). If you have a reason to use a different bootloader, click Change Boot Loader to open the Change Boot Loader dialog box, shown in Figure 3.30.

Note *The terms* bootloader *and* boot loader *are used interchangeably in Red Hat Enterprise Linux.*

FIGURE 3.30
Selecting a bootloader



With Red Hat Enterprise Linux, you can install the GRUB or LILO (Linux Loader) as your bootloader. If you already have a bootloader installed that you don't want to overwrite, select the Do Not Install A Boot Loader option. Make your choice and click OK, or click Cancel to retain the default bootloader. This returns you to the main Boot Loader Configuration screen.

NOTE *You can use GRUB or LILO in concert with another bootloader, such as Partition Magic, System Commander, or Microsoft Windows NTLDR. Choose to install a bootloader, select Configure Advanced Boot Loader Options, and install GRUB or LILO on the partition with the* `/boot` *directory, as described in the next section.*

In the middle of the screen, you can see that Anaconda is installing Red Hat Enterprise Linux in a dual-boot configuration. A Microsoft Windows operating system is installed on partition device `/dev/hda1` labeled DOS, and the main Red Hat Enterprise Linux files are installed on the partition labeled `/dev/hda3`. Red Hat Enterprise Linux is the default, which in the default GRUB configuration means that GRUB starts Linux automatically if you don't make a selection in the GRUB menu within 10 seconds.

You can change a setting associated with DOS or Red Hat Enterprise Linux by highlighting the setting and selecting Edit. This opens the Image dialog box, shown in Figure 3.31, where you can edit the label, change the partition device, and set the associated operating system as the default. Make any desired changes, and click OK to return to the Boot Loader Configuration screen.

You can protect your bootloader with a password. If you want to set a password, click the Use A Boot Loader Password option. This opens the Enter Boot Loader Password dialog box, shown in Figure 3.32, which prompts you to enter a desired password twice. This password keeps others from changing your bootloader configuration file when your computer restarts.

FIGURE 3.31
Bootloader image properties

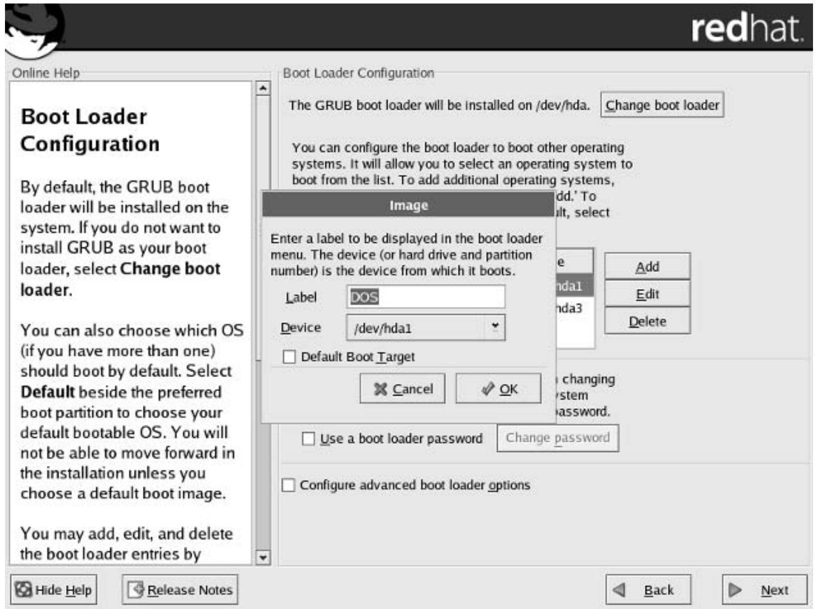
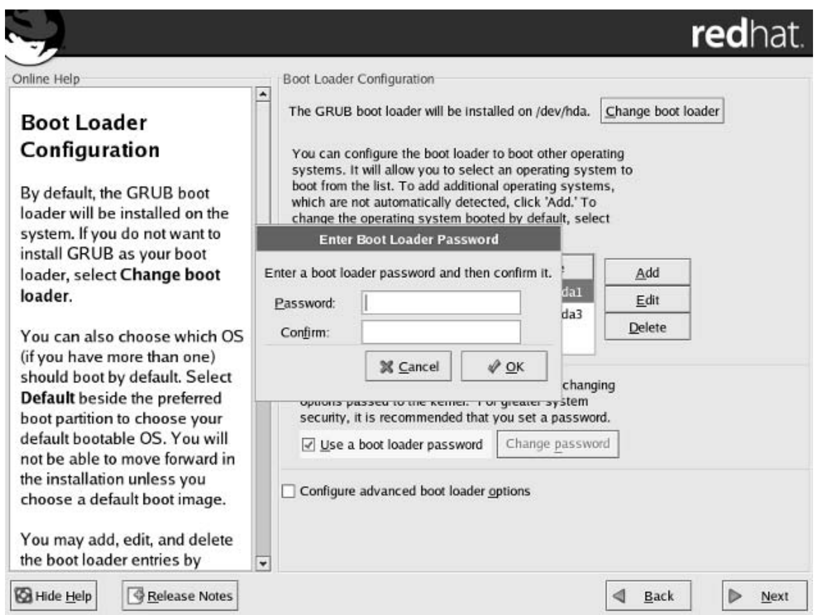


FIGURE 3.32
Enter Boot Loader Password dialog box



Finally, activate the Configure Advanced Boot Loader Options at the bottom of the screen, and click Next to continue.

Advanced Boot Loader Configuration

If you activated Advanced Boot Loader Options, you'll now see the Advanced Boot Loader Configuration screen, shown in Figure 3.33. It allows you to configure several more features associated with your bootloader, as described in Table 3.12. Make any desired changes, and click Next to continue.

Figure 3.33 Advanced Boot Loader Configuration window

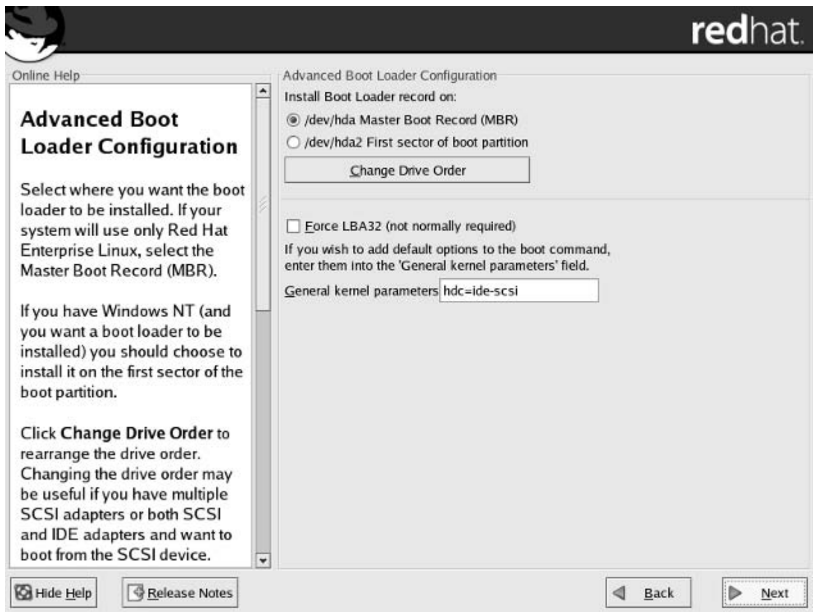


Table 3.12: Advanced Boot Loader Configuration Options

Option	Description
Install Boot Loader Record On	You can configure the bootloader on the MBR of a hard drive, which will run when the BIOS points to that drive. If you have another bootloader on the MBR, you can load the bootloader on the first sector of the boot partition.
Change Drive Order	If you have more than two physical hard drives, you may need to rearrange the drive order to make sure your BIOS looks in the right drive for your bootloader. You can read more about this BIOS hard drive limitation in Chapter 2.

Continued on next page

TABLE 3.12: ADVANCED BOOT LOADER CONFIGURATION OPTIONS *(continued)*

OPTION	DESCRIPTION
Force LBA32 (Not Normally Required)	If you had to mount the `/boot` directory on a partition above the 1024th cylinder on your hard drive, this may help your BIOS find your Linux boot files. This generally isn't required on newer hard drives.
General Kernel Parameters	If you need to pass parameters to the Linux kernel during the boot process, this is a good place to specify them. In the example shown in Figure 3.34, Anaconda added the SCSI emulation module for my CD writer automatically.

NETWORK CONFIGURATION

Now you can configure any network cards detected by Anaconda. By default, network cards are set to automatically get their network parameters from a DHCP server. If you have a DHCP server on your network, it can assign a hostname and give your computer the IP addresses of your network gateway and DNS servers.

Even if you have "just" a home network, you may already have a DHCP server. Many high-speed Internet routers/cable modems/DSL adapters are equipped with a DHCP server. Consult your hardware documentation for information.

Figure 3.34 shows the Network Configuration screen, and Table 3.13 explains the basic options. The two network devices shown in this figure are Ethernet network adapters, `eth0` and `eth1`.

FIGURE 3.34
The Network Configuration window

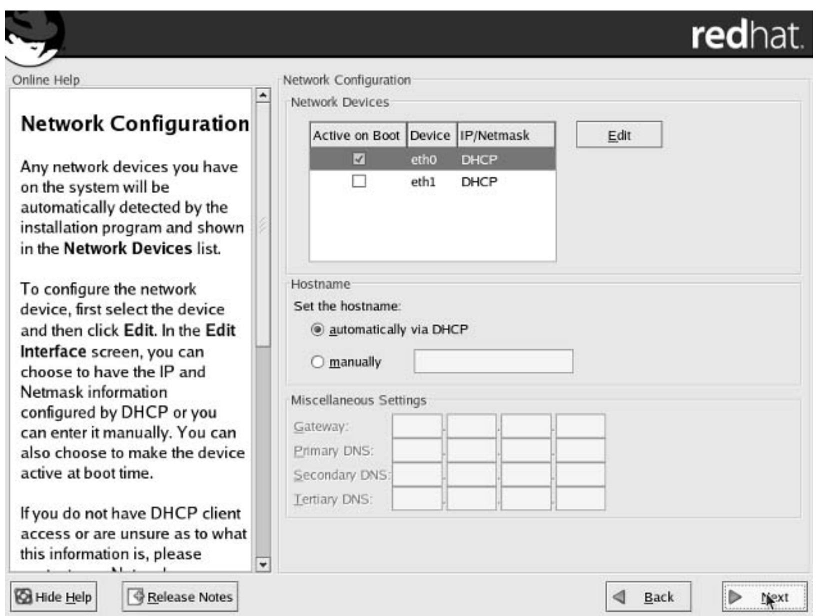


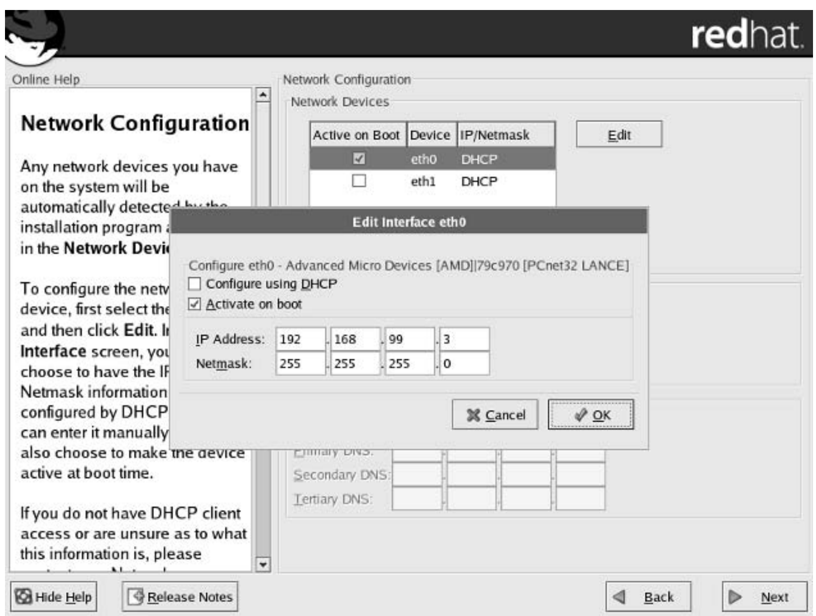
TABLE 3.13: NETWORK CONFIGURATION OPTIONS

Option	Description
Active On Boot	Activates the associated network device during the boot process, if there's a check mark.
Edit	Starts the Edit Interface *Device* window for the highlighted device, where you can configure your network cards manually with a static IP address.
Hostname	Allows you to assign a hostname to this computer; alternatively, a DHCP server can perform this task.
Gateway	Notes the gateway IP address for messages outside your network. You can set it if you've configured your network cards manually.
Primary DNS	Lists the IP address of a DNS server for your network. You can set it if you've configured your network cards manually.
Secondary DNS	Lists the IP address of another DNS server for your network.
Tertiary DNS	Lists the IP address of another DNS server for your network.

If you prefer to assign static IP addresses to your network card, highlight the desired device and click Edit. This opens the Edit Interface *Device* dialog box, shown in Figure 3.35.

To set a static IP address, deselect the Configure Using DHCP option. You can then enter the IP address and netmask of your choice. For guidance on IP addresses on private networks, read Chapter 15. Make any desired changes, and click OK to return to the Network Configuration screen.

FIGURE 3.35
Changing IP address information for a network device



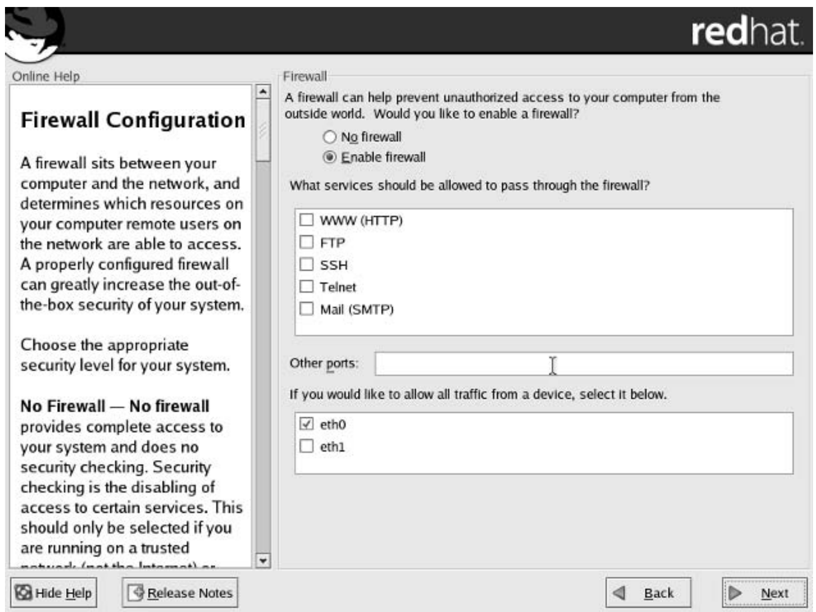
If you change your mind after Red Hat Enterprise Linux is installed, you can edit your configuration with the `redhat-config-network` utility described in Chapter 16.

When you've finished, click Next to continue.

Firewall Configuration

Now you're ready to configure a firewall for Red Hat Enterprise Linux. This is especially important on a gateway computer, which may provide the link between your LAN and the Internet. In that situation, the gateway computer is the best place for a firewall to protect your LAN from the potential ravages of the Internet. For Red Hat Enterprise Linux 3, this creates an `iptables`-based firewall in `/etc/sysconfig/iptables`. Figure 3.36 shows one possible configuration for a gateway computer.

Figure 3.36
Configuring a firewall



The configuration options are as follows.

No Firewall Disables all `iptables` firewall commands on this computer.

Enable Firewall Configures a high security firewall that blocks almost all incoming traffic. The exception is messages from an external DNS server, which supports connections to the Internet.

What Services Should Be Allowed To Pass If you have a server on your computer, you may want to allow incoming traffic from other networks. For example, if you have a web server on your computer, you may want to allow incoming data through the TCP/IP port associated with WWW (HTTP) traffic (see Chapter 25). The other options relate to the File Transfer Protocol (FTP) (see Chapter 22), Secure Shell (SSH) (see Chapter 18), Telnet (see Chapter 18), or a mail server such as sendmail (see Chapter 21).

Other Ports If you want to allow access through your firewall to a different server, you should enter the associated ports and protocols in the associated text box. For example, if you want to allow connections to a secure web server using the HTTPS protocol, you could enter the following in the Other Ports text box:

```
https:tcp,https:udp
```

You can change your firewall settings after Red Hat Enterprise Linux is installed by using the `redhat-config-securitylevel` tool or the `iptables` commands described in Chapter 17. Make any desired changes, and click Next to continue.

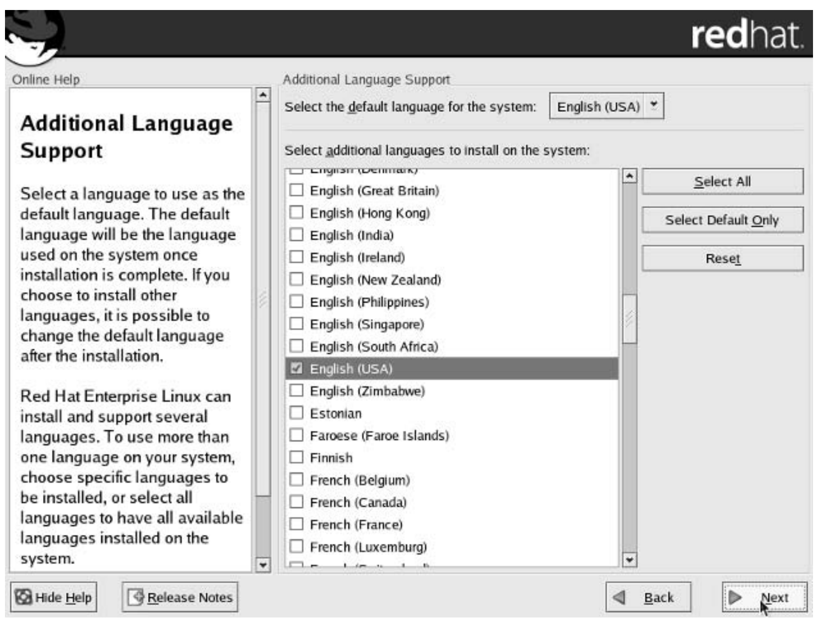
If You Would Like To Allow All Traffic From A Device Lists the network devices on this computer, in this case, `eth0` and `eth1`. It's common to turn off a firewall on the Ethernet card that's connected to the LAN as opposed to one connected to the Internet; in this case, `eth0` is a trusted device.

NOTE *This doesn't affect any firewalls associated with the* `xinetd` *service explained in Chapter 18 or individual network services discussed throughout the book.*

Additional Language Support

In the installation screen shown in Figure 3.37, you can set the default language for Red Hat Enterprise Linux after installation. As you can see, some of the languages include a wide variety of national dialects.

FIGURE 3.37
Selecting languages



If you need different or additional languages for your installation, select them accordingly. If you've configured Anaconda to install more than one language, you can choose the default from these languages by clicking the drop-down arrow adjacent to the Select The Default Language For The System box. You can change the default language after Red Hat Enterprise Linux is installed by using `redhat-config-language`, which is described in Chapter 30. Make any desired changes, and click Next to continue.

Selecting a Time Zone

In this installation screen, you can configure the basic time settings for your computer. After installation, you can go further. If you go through the `firstboot` utility described later in this chapter or `redhat-config-time` in Chapter 13, you can set this computer to synchronize its clock with a central time server.

The Time Zone Selection screen includes two tabs. The Location tab is shown in Figure 3.38.

You can select the time zone associated with your location by clicking on the map or by selecting the location from the scroll window. Unless your computer is in a dual-boot configuration with another operating system such as Microsoft Windows, you should activate the System Clock Uses UTC option. Make your selections, and click the UTC Offset tab, shown in Figure 3.39. UTC is a French acronym that corresponds to Greenwich mean time (GMT).

On the UTC Offset tab, select the offset that matches your time zone; for example, the U.S. West Coast is eight hours behind Greenwich Mean Time, which corresponds to UTC-8. For the United States, you can then activate the Use Daylight Saving Time (US Only) option. Make your choices, and click Next to continue.

Figure 3.38
Setting a time zone location

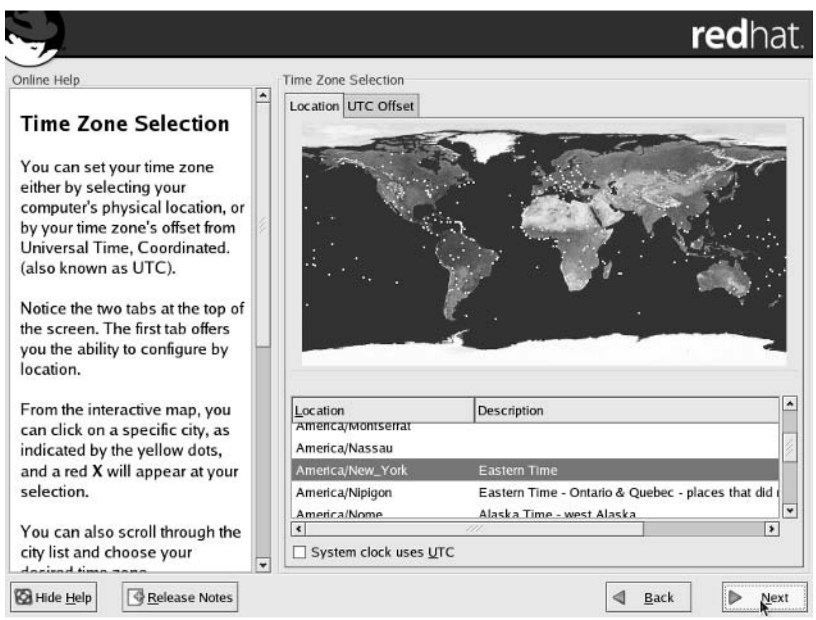
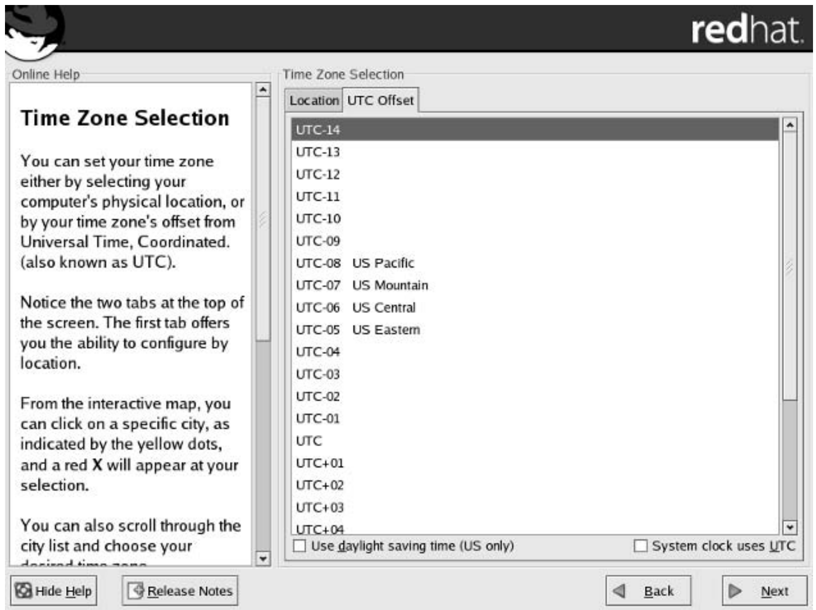


FIGURE 3.39
UTC Offset tab



NOTE *If you're dual booting with another operating system such as Microsoft Windows, don't activate the System Clock Uses UTC option; Windows doesn't know how to handle it.*

Setting a Root Password

The root user is also known as the *superuser*; the root user can do anything on your Linux computer. In the Set Root Password installation screen, shown in Figure 3.40, type your desired root password twice. Red Hat requires the root password that you enter during this process to be at least six alphanumeric characters.

The best passwords include a combination of numbers, uppercase and lowercase letters, and punctuation; it can take days or even weeks for a PC-based cracking program to find that kind of password. Such passwords need not be difficult to remember; I like to create passwords as acronyms for a favorite sentence. For example, `Ieic3teM` could stand for "I eat ice cream 3 times every Monday."

Enter your desired root password twice, and click Next to continue.

Selecting Package Groups

Finally, it's time to select what you're going to install with Red Hat Enterprise Linux. You've configured everything else except your monitor and graphics card. You should now be looking at the Package Installation Defaults screen, shown in Figure 3.41.

FIGURE 3.40
Setting a root password

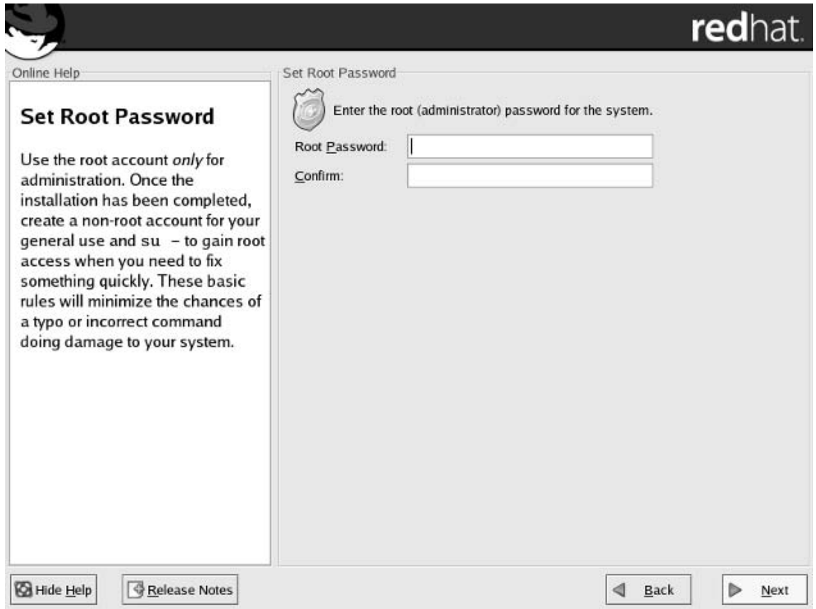
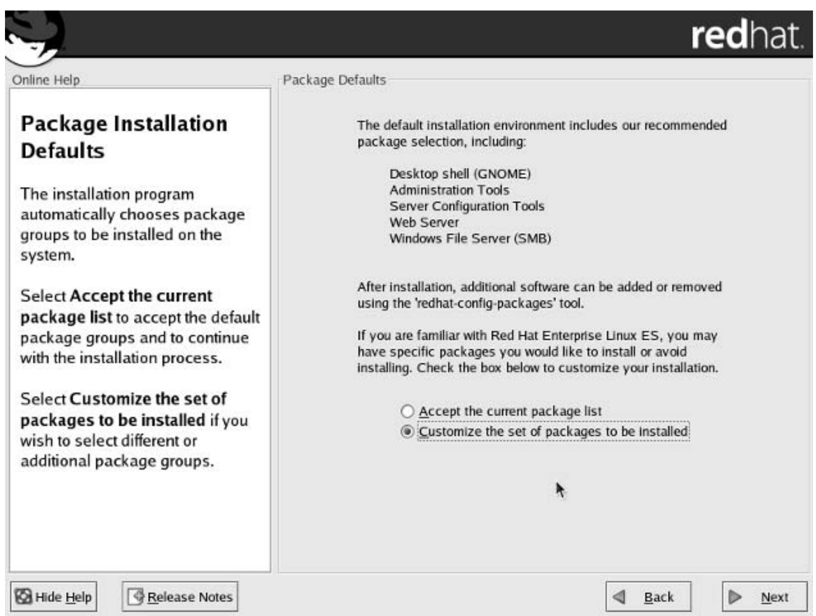


FIGURE 3.41
Default Package Groups



The defaults support a basic Web and Samba (Windows File) server, along with supporting configuration tools as well as the GNOME desktop environment.The defaults are different if you're installing Red Hat Enterprise Linux Workstation. If this is not good enough, activate the Customize The Set Of Packages To Be Installed option and click Next to continue. This brings us to the Package Group Selection installation screen, shown in Figure 3.42.

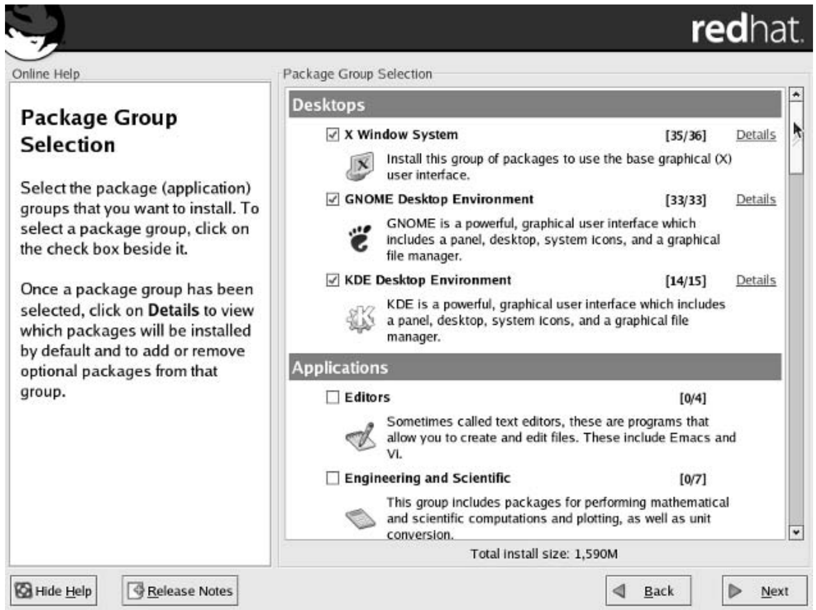


FIGURE 3.42
Selecting package groups

PACKAGES AND GROUPS

If you're unfamiliar with Linux, let's step back a moment. Red Hat organizes software into a package known as an RPM (Red Hat Package Manager). There are approximately 1,100 RPMs on the Red Hat installation CDs. Many of these RPMs depend on each other; for example, you can't use most of the packages associated with the GNOME desktop unless you've also installed the Linux X Window Server.

When you install Red Hat Enterprise Linux, even experienced users don't normally want to pick and choose between 1,100 packages during the installation process. That's one reason why Red Hat has organized the RPMs into package groups displayed in the Package Group Selection installation window.

In other words, an RPM is also known as a *package*, and Red Hat bundles common RPMs together into *package groups*.

The Red Hat package groups correspond to the `comps.xml` configuration file on the first Red Hat Enterprise Linux installation CD, in the `/RedHat/base` directory.

Select the package groups of your choice. If you don't want to install a package group such as Games and Entertainment, you can deselect it to save space for other purposes..

The Anaconda graphical installation organizes package groups into five different categories. There are three desktop groups, as shown in Figure 3.42; Table 3.16 summarizes these groups.

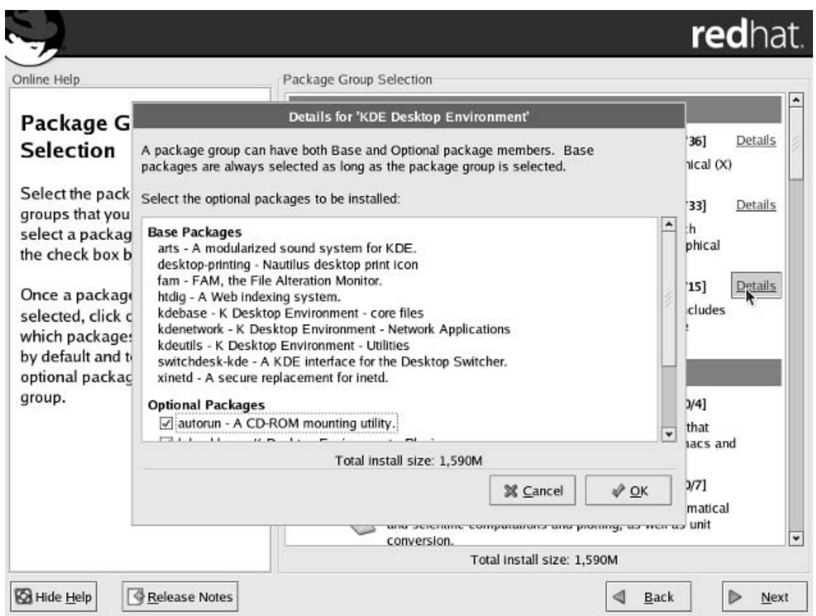
TABLE 3.14: DESKTOP PACKAGE GROUPS

PACKAGE GROUP	**DESCRIPTION**
X Window System	Installs the basic XFree86 server, fonts, and several GUI configuration files
GNOME Desktop Environment	Adds the packages required to use the GNOME Desktop
KDE Desktop Environment	Includes the packages required to use the KDE Desktop

Take a look at the numbers to the right of each package group. For example, in Figure 3.42, look at the numbers associated with the KDE Desktop Environment package group. That tells you that 14 of 15 packages in this package group will be installed. To the right of this number, click Details. This opens the Details For 'KDE Desktop Environment' window, shown in Figure 3.43.

In the Details For 'KDE Desktop Environment' window, packages are organized in three categories: Base, Default, and Optional. Base packages are required for the KDE desktop to work. Default packages are associated with the standard configuration. Optional packages add features.

FIGURE 3.43
KDE Desktop Environment package group details



The next category of package groups is Applications; part of the list is shown in Figure 3.44. Applications range from basic text editors to Internet connection utilities to games. Interestingly enough, this includes the Office/Productivity package group; while it's appropriate for a workstation, it's something that's available on the Red Hat Enterprise Linux 3 server operating systems. The package groups in this category are summarized in Table 3.17.

FIGURE 3.44
Applications package groups

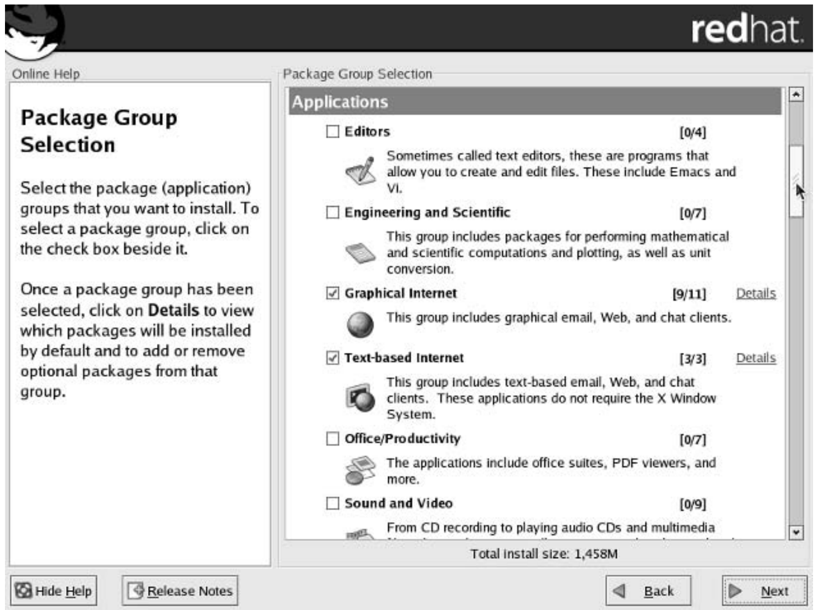


TABLE 3.15: APPLICATIONS PACKAGE GROUPS

PACKAGE GROUP	DESCRIPTION
Editors	Includes the packages for enhanced `vi` and Emacs
Engineering and Scientific	Adds programs for mathematical calculations and graphs
Graphical Internet	Installs a variety of graphical network communication tools
Text-Based Internet	Incorporates network communication tools you can use at the command line
Office/Productivity	Allows you to add a variety of office applications and suites
Sound and Video	Adds a series of multimedia packages, viewers, and configuration tools
Authoring and Publishing	Supports the packages that allow you to create DocBook packages
Graphics	Installs a number of graphical programs and support libraries
Games and Entertainment	Adds various video and board games

One important category for Linux administrators is Servers. Different servers can help you provide services for websites, e-mail, file services, databases, newsgroups, and more. Part of the list is shown in Figure 3.45. Each package group in this category is summarized in Table 3.18. Not all of these package groups are available if you're installing Red Hat Enterprise Linux Workstation.

FIGURE 3.45
Servers package groups

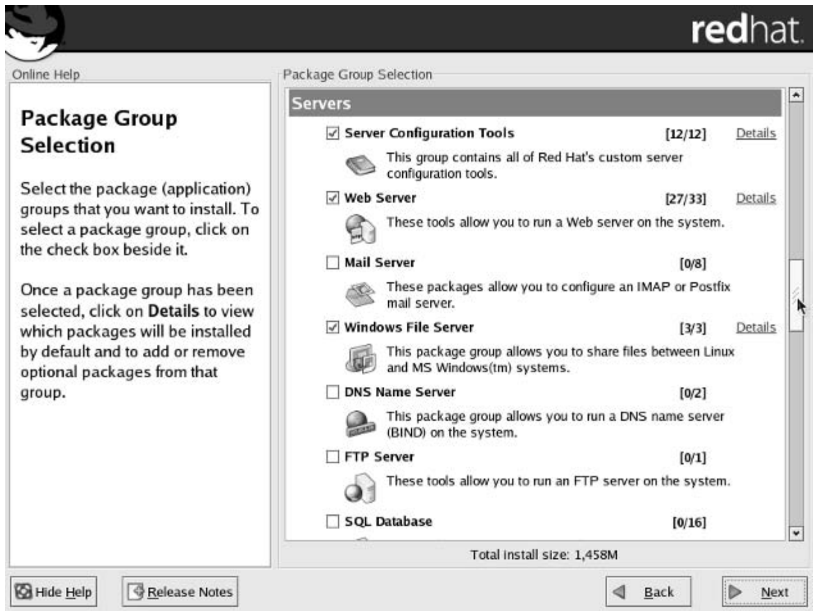


TABLE 3.16: SERVERS PACKAGE GROUPS

PACKAGE GROUP	DESCRIPTION
Server Configuration Tools	Installs several Red Hat graphical configuration tools
Web Server	Adds Apache and related packages for serving web pages to browsers on other computers
Mail Server	Incorporates various e-mail servers and utilities
Windows File Server	Allows you to connect your computer to a Microsoft Windows network as a client and as a server
DNS Name Server	Includes the software required to set up a Domain Name Service (DNS) server or a related caching nameserver
FTP Server	Adds the vsFTP file server, which also supports anonymous access
SQL Database Server	Installs packages that allow you to configure the PostgreSQL server databases
MySQL Database	Installs packages associated with the MySQL server databases
News Server	Adds the InterNetNews package, which supports a Usenet-style newsgroup system
Network Servers	Installs a variety of network servers, including DHCP, quagga, and NIS
Legacy Network Server	Allows you to install older commonly used servers, including RSH and Telnet

There are several Linux development package groups. Even if you're not a developer, you may eventually use many of the packages in these groups. For example, to compile the Linux kernel, you need packages from the Kernel Development and Development Tools package groups. The list of Development package groups is shown in Figure 3.46; Table 3.19 describes each package group in this category.

FIGURE 3.46 Development package groups

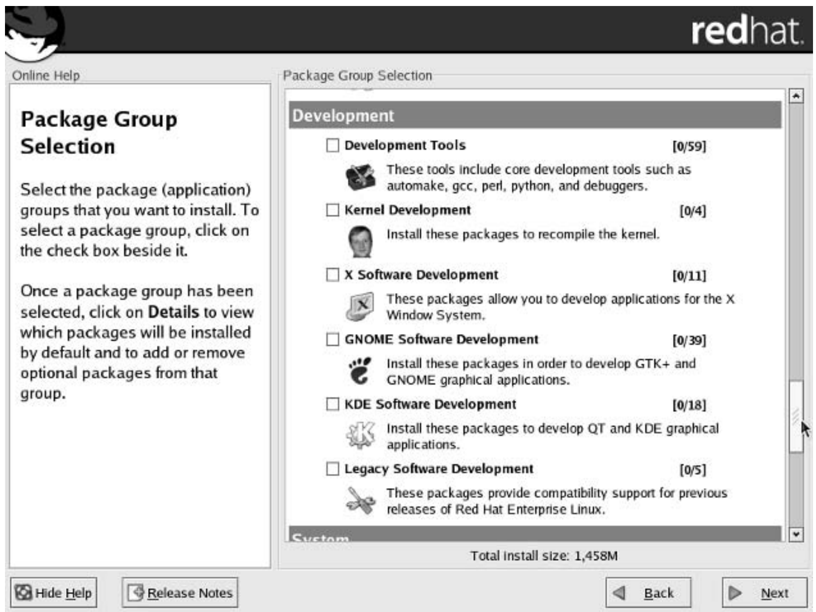


TABLE 3.17: DEVELOPMENT PACKAGE GROUPS

PACKAGE GROUP	**DESCRIPTION**
Development Tools	Includes package tools, language libraries, and more
Kernel Development	Installs headers and kernel source code
X Software Development	Adds development libraries, headers, and documentation associated with the Linux XFree86 graphics system
GNOME Software Development	Incorporates development libraries, headers, include files, and more associated with the GNOME Desktop Environment
KDE Software Development	Incorporates development libraries, headers, include files, and more associated with the KDE Desktop Environment
Legacy Software Development	Supports the use of older C and C++ language libraries

System is the final category of package groups. This category includes administrative, system, and printing tools. The list is shown in Figure 3.47; Table 3.20 summarizes each of the package groups.

FIGURE 3.47
System package groups

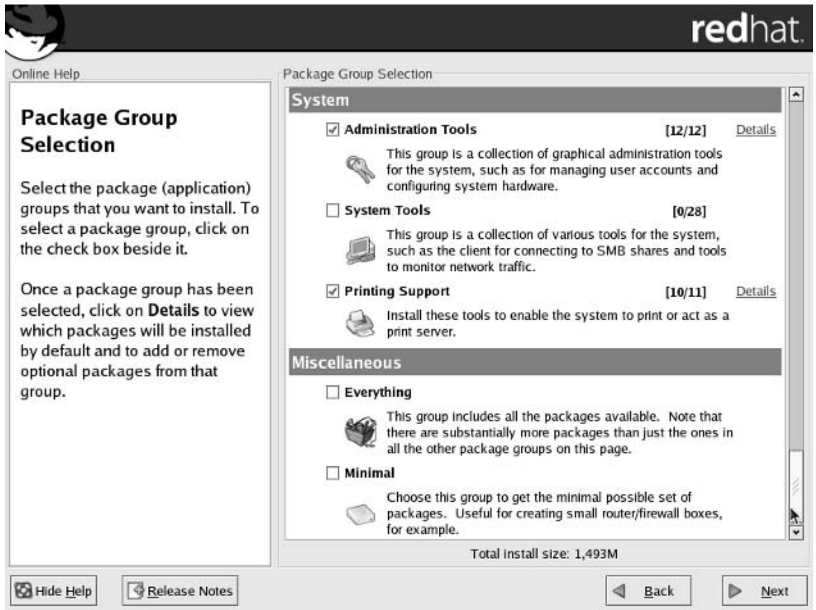


TABLE 3.18: SYSTEM PACKAGE GROUPS

PACKAGE GROUP	DESCRIPTION
Administration Tools	Installs graphical utilities that allow you to administer passwords, packages, kernel parameters, and more
System Tools	Allows you to administer a variety of applications, such as `amanda-client` for backups and `ckermit` for terminal communication
Printing Support	Includes the packages required to install the Common Unix Print System (CUPS)

Finally, at the bottom of the Package Group Selection list, also shown in Figure 3.47, are the following two Miscellaneous options:

Everything Selects all package groups and requires just over 4GB of space just for files.

Minimal Deselects all nonmandatory package groups. After Red Hat Enterprise Linux is installed, you can then install just the packages you need. This is one option that can promote security; in general, if it isn't installed on your computer, it can't serve as a security hole. You need almost 600MB of space for files in this scenario.

NOTE *Anaconda no longer supports the selection of individual RPM packages after this step.*

Ready to Install

Finally, we're ready to let Anaconda install Linux on a computer! Anaconda includes the partitions we've defined, the package groups that we've selected, and the other settings that we've chosen. As you can see in Figure 3.48, when you click Next, Anaconda begins installing Red Hat Enterprise Linux 3 to your specifications.

FIGURE 3.48
Ready to install

Make a note of the listed files. Once installation is complete, you'll be able to review the installed RPMs in `/root/install.log`. The `/root/anaconda-ks.cfg` file can help you duplicate this installation on other computers, using the Kickstart system described in Chapter 5.

When you're ready, click Next to continue.

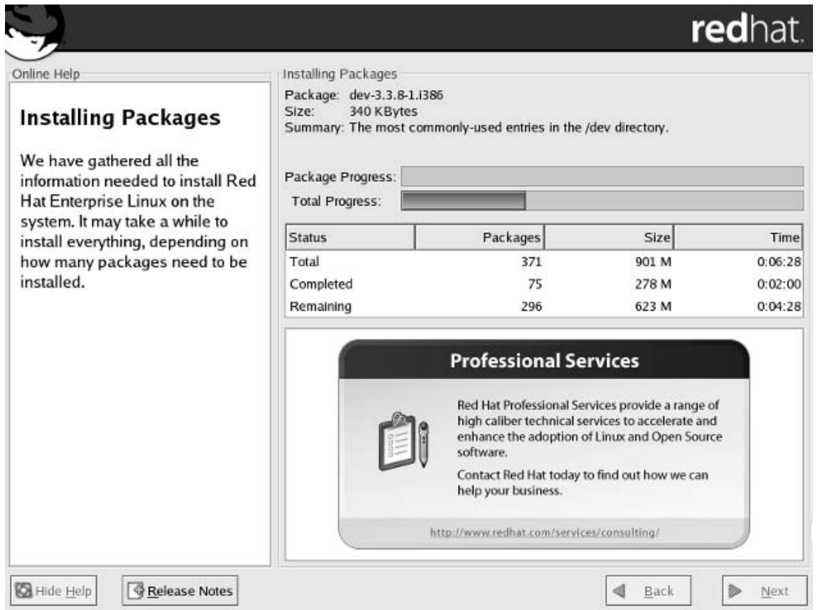
Anaconda Installs Red Hat Enterprise Linux

Finally, Anaconda begins the installation process. First, you'll see a series of messages such as:

```
Formatting / file system...
Formatting /boot file system...
Transferring install image to hard drive...
Transferring updated packages...
Preparing RPM transaction
Starting install process, this may take several minutes
Preparing to install...
```

This is where Anaconda formats the partitions with the selected file system directories. Next, it transfers the basic installation template, as an image, to your hard drive. It sets up the list of RPMs to be installed, and then it starts to transfer data from the installation source—in this case, the Red Hat installation CDs to your hard drive. Then you'll see a screen like the one in Figure 3.49, which constantly updates the progress of the installation.

FIGURE 3.49
The installation in progress



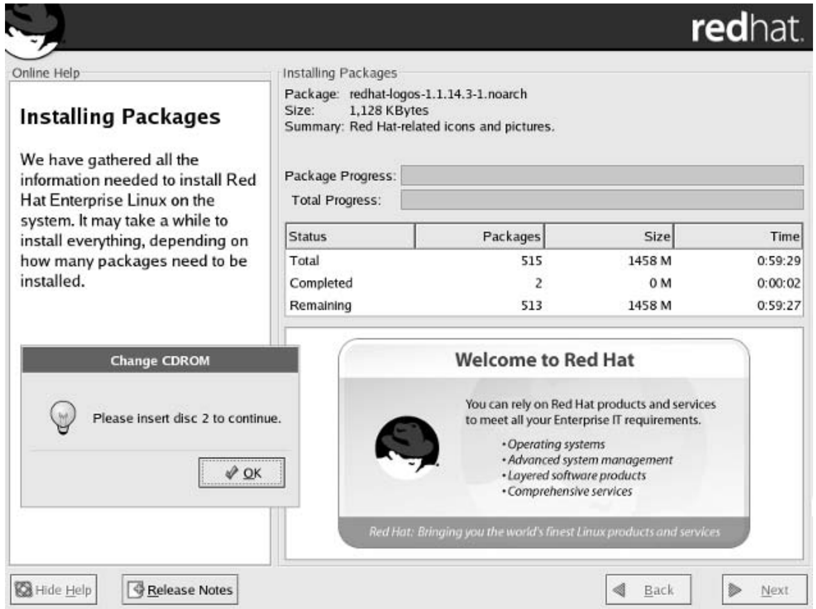
There are four Red Hat Enterprise Linux installation CDs. As the installation progresses, Anaconda may require access to the other CDs, more than once. Installation stops with a message similar to the one shown in Figure 3.50. Follow the instructions and click OK to continue.

This is a good chance to examine what the installation process is doing to your system. Go to the second virtual console. Press Ctrl+Alt+F2. At the command prompt, check for disk usage:

```
-/bin/sh-2.05b# df
Filesystem 1K-blocks    Used Available  Use% Mounted on
rootfs          6120    2471      3299  43% /
/dev/root.old   6120    2471      3299  43% /
/tmp/cdrom      462464  462464       0   100% /mnt/source
/tmp/sda2       3771316 32844  3546900   1% /mnt/sysimage
/tmp/sda1       101089  4127     91743   5% /mnt/sysimage/boot
```

This output tells us that Anaconda has mounted the root (/) directory partition on the `/mnt/sysimage` directory. It has also mounted the partition with the `/boot` directory on `/mnt/sysimage/boot`.

FIGURE 3.50
Time for another CD



You can use bash shell commands (described in Chapters 6, 7, and 8) to navigate these directories to see what Anaconda has installed so far. In fact, if there's a problem, examine the contents of `/mnt/sysimage/root/install.log`. This log identifies the current RPM that Anaconda is attempting to install on your system. If your installation freezes, there may be a problem with that particular RPM on your CD (or network installation server).

Once all desired RPM packages are installed, you'll see the following messages, which get Anaconda ready for the next steps in the process and installs the bootloader on your MBR. *At this time,* you can also find the bootloader configuration file in the second virtual console in the `/mnt/sysimage/etc` directory. Once you've completed your installation, you can find the file in the expected location, in the `/etc` directory. If you're using GRUB, it's in the `grub.conf` file.

```
Performing post install configuration
Installing bootloader
```

Managing Post-Installation Steps

Anaconda has installed Red Hat Enterprise Linux on your computer. But your work isn't done. You still need to configure your graphics systems. The next screen you should see is shown in Figure 3.51.

NOTE *Red Hat Enterprise Linux 3 doesn't support the creation of a boot floppy during the installation process; however, it's easy to create with the* `mkbootdisk` *command described in Chapter 11.*

Configuring a Video Card

If you've installed X Window software, Anaconda now prompts you to configure your graphics system, as shown in Figure 3.51. You can skip this process and configure it later with the `redhat-config-xfree86` utility described in Chapter 29.

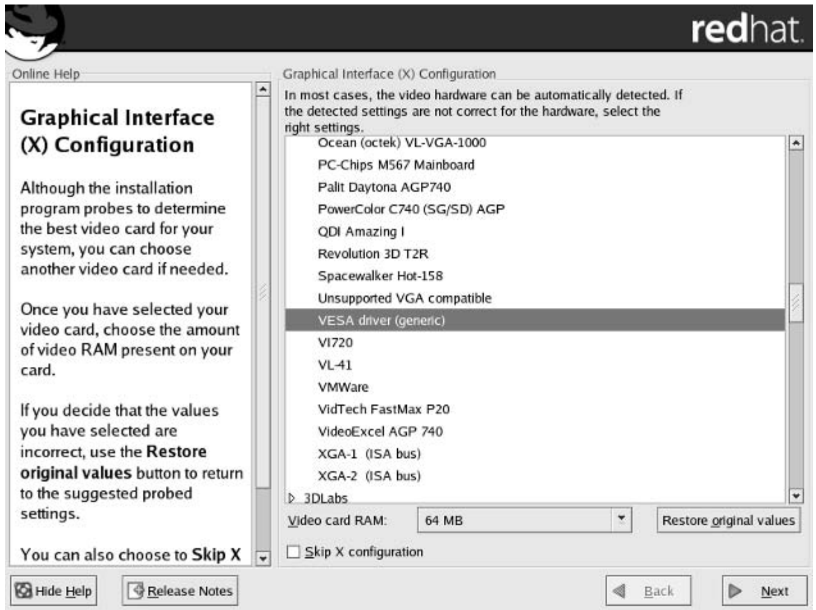


Figure 3.51 Configuring the Linux graphical interface

If Anaconda detected your hardware at the start of this process, it highlights the graphics card it detected, along with the amount of RAM on that card. The options in the Graphical Interface (X) Configuration screen are as follows:

Video card Select the video card that most closely matches your hardware. Anaconda may have selected one for you. Cards are organized by manufacturer. If you don't see your card, it may be under the manufacturer labeled Other at the top of the list. Alternatively, almost all newer video cards can be configured as a generic video card.

This section includes three truly generic cards: Generic VGA Compatible; Unsupported VGA Compatible; and VESA Driver (Generic), which is equivalent to SVGA.

Note *The Video Graphics Adapter (VGA) standard is a standard associated with older graphics cards and a monitor resolution of 640×480. SVGA stands for Super VGA, and is associated with a resolution of 800×600. These standards are maintained by the Video Electronics Standards Association (VESA); the standard VESA driver is associated with SVGA video cards. Many unrecognized high-performance cards such as those that conform to XGA and SXGA standards can use VESA mode.*

Video Card RAM Set the RAM to the capacity of your video card. If your video card shares regular RAM, make sure this matches the associated setting in your BIOS. Anaconda allows you to set your video RAM in increments between 512KB and 128MB.

Restore Original Values If you've made a number of changes and want to return to the original detected configuration, click this button.

Skip X Configuration If you don't want to configure your graphics system at this time, enable this option and click Next. You can still configure your graphics system later with `redhat-config-xfree86`.

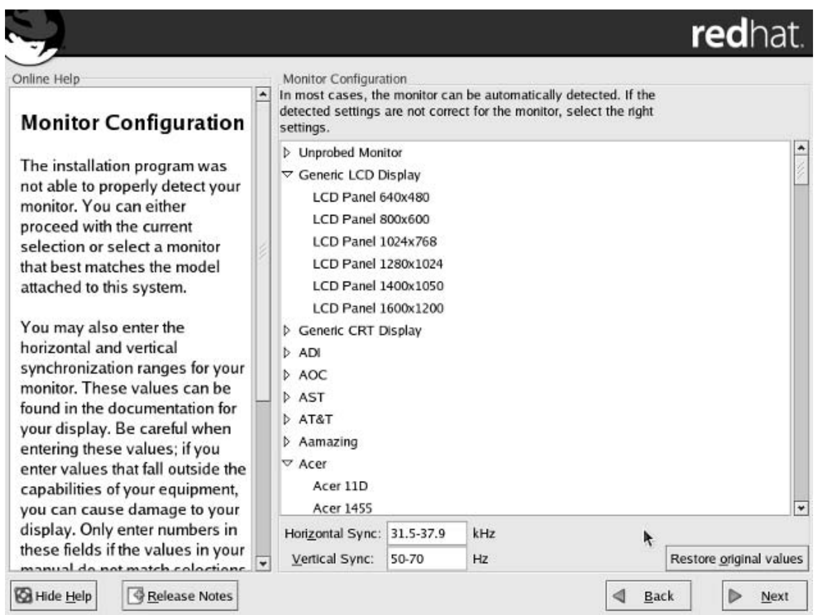
In most cases, you won't need to make any changes. Make any desired changes, and click Next to continue.

Configuring a Monitor

If you still want to configure the X Window system on your computer, the next screen you'll see should be similar to the Monitor Configuration screen shown in Figure 3.52.

When Anaconda probed your system, it may have detected a monitor. If it did, you'll see it highlighted here. If this is the wrong monitor, select your monitor from the list, which is classified by manufacturer and model. If you don't see your monitor on the list, you can select from a wide variety of generic monitors.

Figure 3.52
Monitor Configuration screen



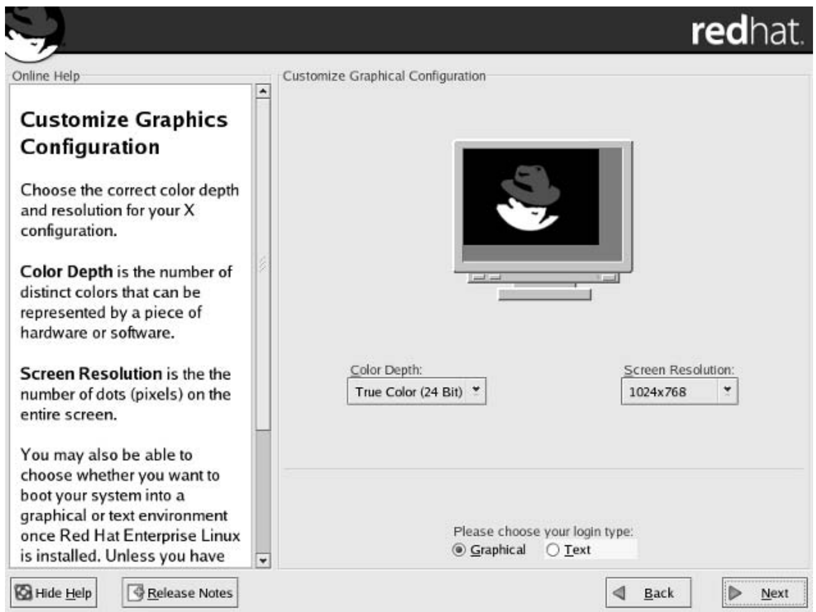
Warning *Every monitor has a horizontal sync and vertical sync rate. Check the documentation for your monitor carefully! If the numbers you set here exceed the capabilities of your monitor, the signals from your video card could blow out your monitor's circuitry.*

If you experiment and would rather return to the values detected by Anaconda, click Restore Original Values. In most cases, you won't need to make any changes. Make any desired changes, and click Next to continue.

Customizing Graphics

We've arrived at the last step! Now you get to put together the configuration settings for your video card and monitor. Figure 3.53 shows the Customizing Graphics Configuration screen.

Figure 3.53 Customizing your graphics configuration



The options in this screen are as follows:

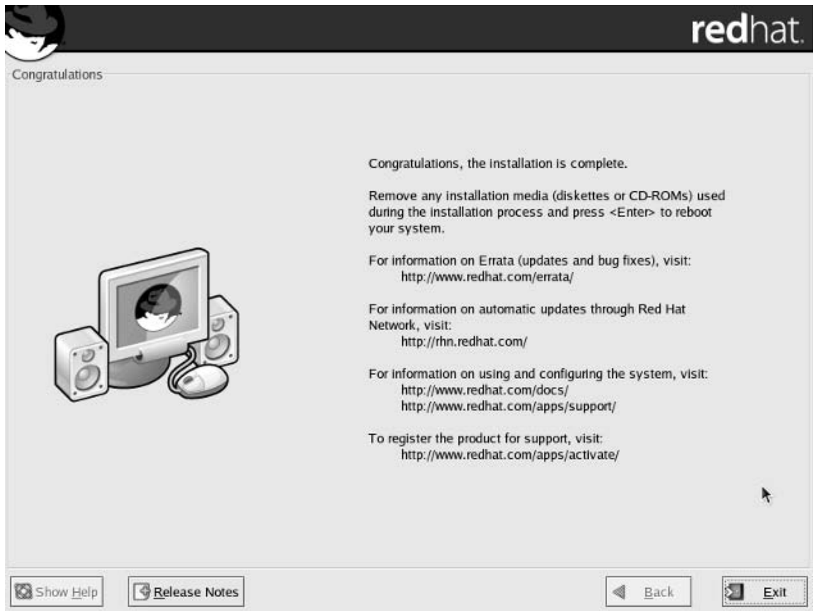
Color Depth Specifies the number of bits of color associated with each pixel. For example, 24 bits is "true color" because it supports rendering of up to 2^{24}, which equals 16,777,216 different colors. Depending on the capabilities of your video card and monitor, you may be able to set 8 bits (256 colors) or 16 bits (65,536 colors) for each pixel.

Screen Resolution Sets the number of pixels to be displayed on your monitor. The screen resolution is set in a `horizontal` × `vertical` format. For example, 800 × 600 resolution represents 800 pixels in the horizontal dimension and 600 pixels in the vertical dimension on your monitor. Available resolutions depend on the size of the monitor and the RAM associated with your video card.

Please Choose Your Login Type Configures the Linux boot sequence; this affects the `id` variable in `/etc/inittab`, which you can change as described in Chapter 11. If you select Graphical, Linux boots into a graphical login screen; if you select Text, Linux allows you to log in at a text-based virtual console.

Make your selections, and click Next to continue. Finally, the installation is complete, as shown in Figure 3.54. Click Exit to reboot your computer.

FIGURE 3.54
Installation is complete!



Running the Red Hat Setup Agent

The installation process may seem long enough already. Red Hat has moved several configuration activities from installation to a new program known as the Red Hat Setup Agent, also known as `firstboot`.

The first time you reboot your computer, you should see your chosen bootloader. By default, the bootloader is GRUB, which is shown in Figure 3.55.

As described near the end of the installation process, there are two possible login modes: text and graphical. A typical text login looks like the following:

```
Red Hat Enterprise Linux ES release 3 (Taroon)
Kernel 2.4.21-4.EL on an i686

Enterprise3 login:
```

However, if you selected a graphical login type at the end of the installation process, you'll be taken to the Red Hat Setup Agent, shown in Figure 3.56. The Red Hat Enterprise Linux boot process won't allow any detours before you're allowed to log in at a graphical screen.

FIGURE 3.55
The GRUB bootloader

FIGURE 3.56
Red Hat Setup Agent

The Red Hat Setup Agent allows you to configure user accounts, set up connections to a time server, probe for a sound card, register with the Red Hat Network, and add extra software.

If you selected a text login, you can start the Red Hat Setup Agent with the `firstboot` command.

NOTE *If you selected a text-mode login, you're in runlevel 3. Run the* `telinit 5` *command and log into the Linux GUI. You can then run* `firstboot` *in a GUI command-line interface.*

Next, you'll get to review the Red Hat Enterprise Linux 3 license agreement, as shown in Figure 3.57. You can find a full copy of the agreement at `www.redhat.com/licenses`. It varies somewhat by country. While I'm not a lawyer, you can read for yourself how the Red Hat (U.S.) agreement respects the Linux General Public License (GPL) status of most of the software, as shown in Appendix B. However, Red Hat is the only company that can share or sell its distribution with its trademarks, which include its name and logos.

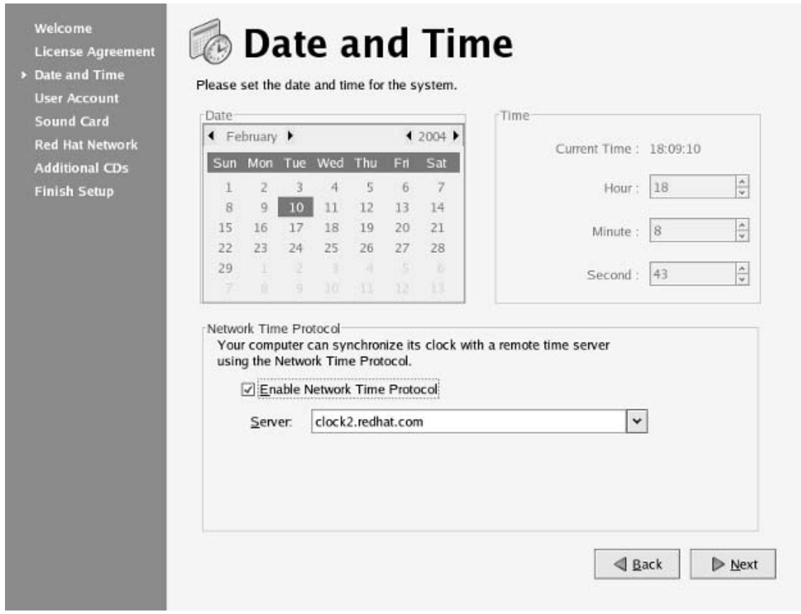
FIGURE 3.57
The Red Hat license agreement

Select Yes, I Agree To The License Agreement, and click Next to continue. Otherwise you're prompted to shut down your computer and remove Red Hat Enterprise Linux 3 from your system.

Specifying a Date and Time

Yes, you already specified a date and time during the Red Hat Enterprise Linux installation process. The difference here is that this `firstboot` screen allows you to synchronize your computer with a central time server. Place a check mark in the Enable Network Time Protocol option, as shown in Figure 3.58.

FIGURE 3.58
Specifying a time server



If your computer is connected to the Internet, you may want to select an NTP server from what may be the authoritative website on NTP, `www.ntp.org`; it includes a link to a list of active NTP servers around the world. You can change your settings later with the `redhat-config-time` utility described in Chapter 13. This includes two standard Red Hat time servers (`clock.redhat.com` and `clock2.redhat.com`).

To make sure that that the NTP daemon continues working the next time you start Linux, the Red Hat Setup Agent activates the NTP daemon, `ntpd`, at runlevels 3 and 5. However, it won't work if you've set up a firewall during the installation process (unless you've specifically allowed traffic through port 123). For more information on runlevels, read Chapter 11. For more information on firewalls, read Chapter 17.

Specify the time server of your choice. If you're connected to the Internet, `firstboot` now tries to contact your selected time server.

Creating a Regular User

You're encouraged to create a personal user account, as shown in Figure 3.59. Enter a login name in the Username text box. Then add identifying information in the Full Name text box. Enter the same password twice in the last two text boxes (Vaclav Havel is the recently departed president of the Czech Republic).

The account can be part of a network service such as NIS, LDAP, or even a Microsoft account via Samba. The Use Network Login option opens the Authentication Configuration window, where this can be configured. I describe this tool in more detail in Chapter 23. Add the user of your choice.

FIGURE 3.59
Creating a regular user

Detecting a Sound Card

The Red Hat Setup Agent automatically tries to detect any sound cards that may be located on your computer. If it succeeds, you'll see a sound card vendor, model, and module, as shown in Figure 3.60. If you have speakers connected to your sound card, you can click the Play Test Sound button.

Once `firstboot` finishes playing the sound, you'll see the prompt asking "Did you hear the sample sound?" If you didn't, click No, and you'll get a message telling you that the sound card wasn't activated.

Registering with the Red Hat Network

When you register your computer with the Red Hat Network, you can set up your computer to receive the latest software upgrades and patches. You don't have to register immediately, as shown in Figure 3.61.

A connection to the Red Hat network requires a subscription, and is part of the cost of an official copy of Red Hat Enterprise Linux. As you can see, you don't have to connect your computer immediately. You'll first need to set up your account through `rhn.redhat.com`. We guide you through the process in Chapter 10. If you want to set up your computer now, refer to that chapter for guidance. For the purpose of this chapter, select No, I Do Not Want To Register My System, and click next to continue.

FIGURE 3.60 `firstboot` detects a sound card.

FIGURE 3.61 Basic Red Hat network services

Additional Installation

If you want to install additional packages in Red Hat Enterprise Linux, this is your chance. As shown in Figure 3.62, you can install additional packages from the Red Hat Enterprise Linux Documentation CD, the Red Hat Enterprise Linux Installation CD, or Additional CDs.

FIGURE 3.62
Ready to install more

TIP *Unfortunately, this step in the First Boot process works only if you run* `firstboot` *after you log into this computer. As of this writing, this is a documented problem per* `bugzilla.redhat.com/bugzilla/show_bug.cgi?id=106087`.

Insert the appropriate CD, click Install, and then follow the prompts. This section uses the software associated with the `redhat-config-packages` utility described in Chapter 10 to organize the installation of new software. Follow the prompts, and `firstboot` automatically installs the desired documents from CD.

If you want to add more RPMs from the installation CDs, insert the first Red Hat installation CD and click Install. This starts the `redhat-config-packages` utility.

At this point, you should see the Finish Setup screen, shown in Figure 3.63. As noted, your system is now ready to set up and use.

FIGURE 3.63
Setup is finished.

Troubleshooting the Installation

When you're troubleshooting a problem, the scientific method suggests that you first gather all available data. During the installation process, you can obtain a lot of troubleshooting data through the virtual consoles. Once you've done this work, you can identify the symptoms of the problem and use your support options with Red Hat, pose your question on one of the related mailing lists, or address the problem to the Linux community through your local user group or online.

We describe some typical problems in the following section.

Installation Virtual Consoles

One of the key tools for troubleshooting a problem installation is the *virtual consoles*. Once the graphical installation process begins, you can access five different installation virtual consoles.

When you're having a problem with installation of Red Hat Enterprise Linux, the problem may not be obvious. Several text installation screens can provide valuable messages. You can get to these screens with the Ctrl+Alt+F*n* command, where *n* is the virtual console number: 1, 2, 3, 4, 5, or 7.

Once you've reviewed the messages, you can return to the installation screen with the Ctrl+Alt+F7 command. Table 3.19 describes the installation screens.

TABLE 3.19: RED HAT INSTALLATION SCREENS

Screen	Description
Ctrl+Alt+F1	Looks at the detection messages for the local video card, monitor, and mouse; view of installation screens if you're installing in text mode.
Ctrl+Alt+F2	Opens a bash shell with limited command capabilities; for example, the `df` command can show mounted directories and partitions. Other bash commands are described in Chapters 6, 7, and 8 of this book.
Ctrl+Alt+F3	Views the installation log, with messages related to hardware detection; trouble reading CDs or loading drivers may be found here. During the installation process, this information is recorded in `/tmp/anaconda.log`.
Ctrl+Alt+F4	Goes to the system message log, with messages such as formatting and mounting directories on partitions. During the installation process, this information is recorded in `/tmp/syslog`.
Ctrl+Alt+F5	Notes other messages, such as filesystem labels, blocks, formats, and journals. Accessible only after Anaconda formats partitions.
Ctrl+Alt+F7	Returns to the graphical installation screen.

NOTE *When changing screens during the installation process, some keyboards require that you use the Ctrl and Alt keys on the left side of the keyboard.*

Installation virtual consoles and log files in the `/tmp` directory are stored in a RAM disk; thus, they're deleted once you reboot your computer or finish the installation process.

Graphics-Detection Messages

Early in this chapter, we reviewed messages in the first console associated with a successful installation. But problems are possible, especially if you have non-conforming graphics hardware. First, let's take a look at a message on my laptop that does not have enough memory:

```
You do not have enough RAM to use the graphical installer. Starting text mode.
```

This message is straightforward; if you see it, you need a computer with additional memory to perform a graphical installation. If you have a computer with the minimum RAM supported by Red Hat, this shouldn't be a problem. Fortunately, text-mode installation (covered in Chapter 4) is sufficient for most purposes. Sometimes graphics hardware doesn't conform, as indicated by the following messages:

```
Running anaconda, the Red Hat Enterprise Linux system installer - please wait...
Probing for video card: Unsupported VGA Compatible
```

```
Probing for monitor type: Unknown monitor
Probing for mouse type: Generic - Wheel Mouse (PS/2)
Attempting to start native X Server
Waiting for X server to start...log located in /tmp/X.log
1...2...3...4...5...X SERVER FAILEDAttempting to start VESA driver X server X
startup failed, falling back to text mode
```

These messages are also fairly straightforward, suggesting that this computer doesn't include graphics hardware that conforms even to the VESA (SVGA) standard.

Sometimes the graphics card and monitor, as detected by Anaconda, aren't compatible. If you need a graphical installation, you can try to force a lower setting, something that's usually easier to handle for most hardware. For example, the following command at the first installation `boot:` prompt tries to set up Anaconda in a minimal graphical environment:

```
boot: linux resolution=640x480
```

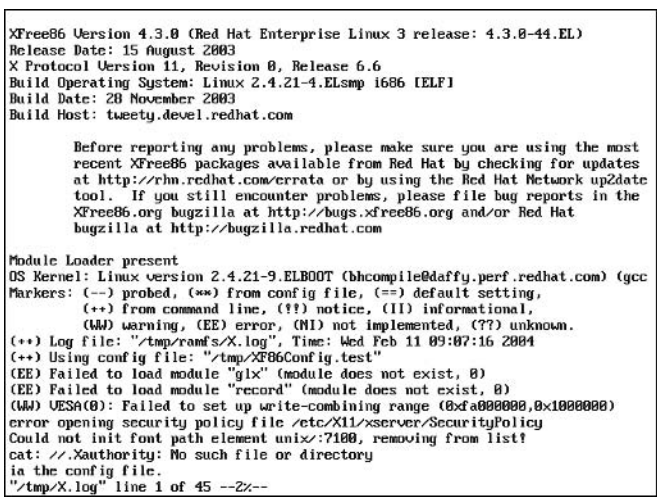
Log Files

We have surprisingly easy access to log files during the installation process, through the second virtual console. Press Ctrl+Alt+F2 to open a bash prompt:

```
-/bin/sh-2.05b#
```

Here you can enter the bash commands of your choice. Any files installed so far are accessible through this interface. Earlier, we saw the message for the `/tmp/X.log` file. Open it with the `vi /tmp/X.log` command. The file should look similar to Figure 3.64.

Figure 3.64
An X Configuration log



```
XFree86 Version 4.3.0 (Red Hat Enterprise Linux 3 release: 4.3.0-44.EL)
Release Date: 15 August 2003
X Protocol Version 11, Revision 0, Release 6.6
Build Operating System: Linux 2.4.21-4.ELsmp i686 [ELF]
Build Date: 28 November 2003
Build Host: tweety.devel.redhat.com

        Before reporting any problems, please make sure you are using the most
        recent XFree86 packages available from Red Hat by checking for updates
        at http://rhn.redhat.com/errata or by using the Red Hat Network up2date
        tool.  If you still encounter problems, please file bug reports in the
        XFree86.org bugzilla at http://bugs.xfree86.org and/or Red Hat
        bugzilla at http://bugzilla.redhat.com

Module Loader present
OS Kernel: Linux version 2.4.21-9.ELBOOT (bhcompile@daffy.perf.redhat.com) (gcc
Markers: (--) probed, (**) from config file, (==) default setting,
         (++) from command line, (!!) notice, (II) informational,
         (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(++) Log file: "/tmp/ramfs/X.log", Time: Wed Feb 11 09:07:16 2004
(++) Using config file: "/tmp/XF86Config.test"
(EE) Failed to load module "glx" (module does not exist, 0)
(EE) Failed to load module "record" (module does not exist, 0)
(WW) VESA(0): Failed to set up write-combining range (0xfa000000,0x1000000)
error opening security policy file /etc/X11/xserver/SecurityPolicy
Could not init font path element unix/:7100, removing from list!
cat: //.Xauthority: No such file or directory
ia the config file.
"/tmp/X.log" line 1 of 45 --2%--
```

Note the comments at the bottom of the file, pointing you to `/tmp/ramfs/X.log`, which provides additional information about the graphics problem on this computer. Other important log files are readily available in the `/tmp` directory, as explained in Table 3.20.

Table 3.20: Log Files during the Installation Process

File	Description
`anaconda.log`	Hardware-detection log associated with the third virtual console
`isoinfo`	MD5 checksum for the current CD
`modules.conf`	List of installed modules
`syslog`	Boot log; corresponds to `dmesg` (see Chapter 11)
`X.log`	Graphical configuration log file
`ramfs/X.log`	Detailed graphical configuration log
`XF86Config.text`	Preliminary X Window configuration file

Hardware-Detection Messages

Several hardware-detection messages are available in the third virtual console. During the installation process, you can get to this console with the Ctrl+Alt+F3 command, or you can see the entire list of messages in the second virtual console in `/tmp/anaconda.log`. Just remember, as installation proceeds, Anaconda constantly adds information to this file.

If you're having a hardware problem, it will normally be fairly obvious; for example, the following message indicates a problem that Anaconda has reading one of my CD-ROM drives:

```
<4>hdb: cdrom_decode_status: error=0x51{DriveReady SeekComplete Error}
```

While this message could indicate a problem with the CD media or hardware, it does tend to identify the problem.

Sometimes hardware messages are subtler.

```
/tmp/yenta_socket.o: init_module
Hint: insmod errors can be caused by incorrect module parameters, including
 invalid IO or IRQ parameters.
You may find more information in syslog or the output from dmesg.
```

I knew that the `yenta_socket.o` module is related to my PCMCIA hardware; it took additional research to find that my boot disk was missing the i82365 PCMCIA module. It's like the dog that didn't bark; I didn't figure out the problem until I realized that Anaconda never loaded the key PCMCIA module. I wouldn't have figured that out had I not been familiar with the hardware on my laptop.

One more common error is a signal 11, also known as a *segmentation fault*. This generally indicates a hardware problem. For example, if I lose a connection from a VMware installation to the CD drive, I end up with a signal 11 and am prompted to reboot to start the installation process all over again. Other possible causes of this error are the CPU cache in the BIOS and unrecognized RAM. For

example, you can try downgrading RAM available at the first installation `boot:` prompt with the following command:

```
boot: linux mem=256M
```

The System Message Log

The standard Linux installation message log is filled with fairly standard boot messages. It's less likely you'll see a problem here. For example, any hardware that isn't detected simply doesn't show up in the system message log.

Thus, in order to find problems through this log, you need to be a bit of a detective. For example, you know there's a problem if you see a message detecting only 256MB of memory when you have 512MB installed.

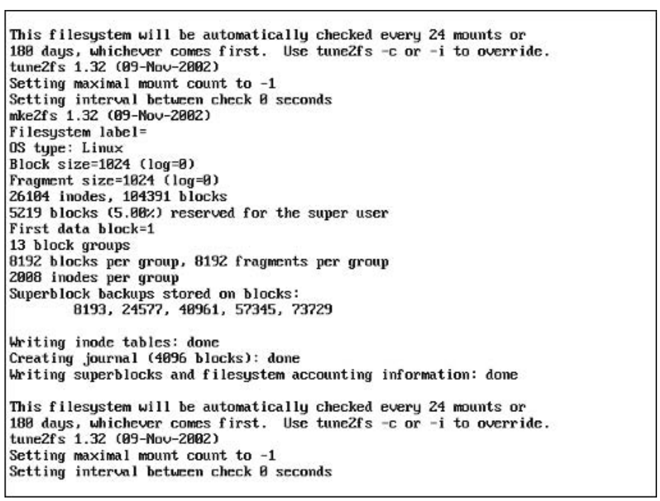
This log is associated with the fourth installation virtual console, which you can access with the Ctrl+Alt+F4 command. You can also review the messages from the second virtual console in the `/tmp/syslog` file. Keep in mind that, as installation proceeds, Anaconda constantly adds information to this file.

Other Messages

Anaconda formats your partitions just before it actually starts to install Red Hat Enterprise Linux. If you haven't configured partitions with sufficient space, you'll get an error message and will have to start the process again.

You can take a look at this console after Red Hat Enterprise Linux starts to install packages on your computer by using the Ctrl+Alt+F5 command. We've shown a view in Figure 3.65, which includes messages on how Anaconda has formatted the root (`/`) directory filesystem.

Figure 3.65
Anaconda format messages



```
This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
tune2fs 1.32 (09-Nov-2002)
Setting maximal mount count to -1
Setting interval between check 0 seconds
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
26104 inodes, 104391 blocks
5219 blocks (5.00%) reserved for the super user
First data block=1
13 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
tune2fs 1.32 (09-Nov-2002)
Setting maximal mount count to -1
Setting interval between check 0 seconds
```

Later in this process, Anaconda presents a GRUB prompt that lets you modify your bootloader configuration. However, the GRUB configuration file, `grub.conf`, is accessible through the second virtual console, as described earlier in the "Anaconda Installs Red Hat Enterprise Linux" section.

Package Status

One all too common problem with Linux installations is an RPM package that wasn't copied correctly. It could be the 1,000th package in the installation process. If suddenly Anaconda finds a problem with a specific package, the installation stops. Unless you have alternate media (such as duplicate CDs) at hand, you may have no recourse but to restart the installation.

Once installation proceeds, you can track the status of the installation on the screen. Both graphics- and text-mode installations identify the package currently being installed. There is one more source; once installation starts, you can find the current list through the second virtual console, in the `install.log` file located in the `/mnt/sysimage/root` directory.

If you can identify the package with the problem, you may be able to replace it. You could replace it in the list of packages on the CD, or if you're more fortunate, you could download the package again to a central network installation source.

Especially if you've downloaded your Red Hat installation CDs over the Internet, there are many possible causes. There could be a momentary power surge somewhere on the Internet. You could be downloading to a hard drive with a bad sector. You may copy the CD files onto a disk with a flaw. The possible causes go on and on. Although installing Linux from downloaded CDs is usually trouble free (I do it all the time), it does have its share of risks.

Logging In

Now you and your computer are ready for Linux. If you're a Linux expert (or want to be), you're probably logging in from the command-line interface, as shown here:

```
Red Hat Enterprise Linux ES release 3 (Taroon)
Kernel 2.4.21-4.EL on an i686

Enterprise3 login: username
Password:
Last login: Wed Mar 19 15:33:00 on tty1
[username@Enterprise3 username]$
```

Now you're ready for a command-line interface, which is the main focus of most of this book.

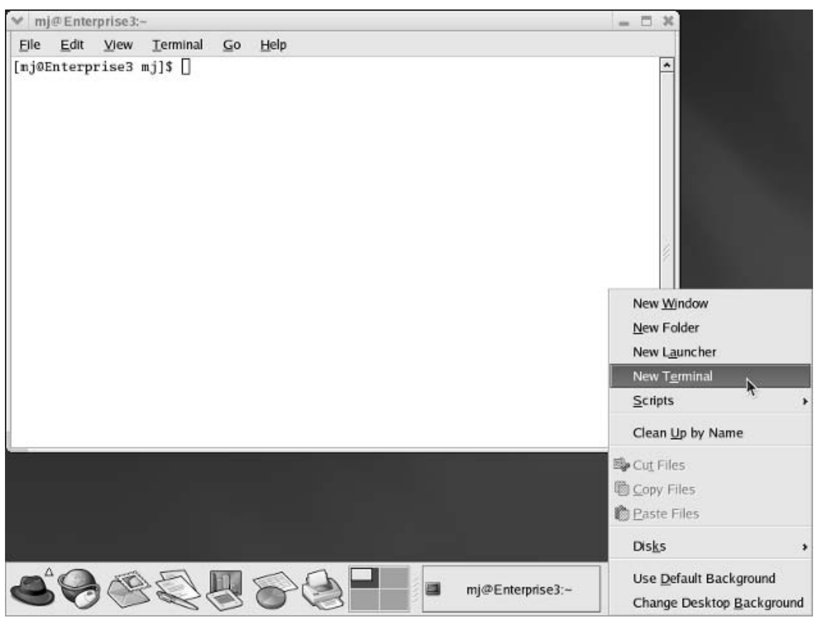
Alternatively, if Linux is relatively new to you, you may be logging in at a graphical login prompt, such as the one shown in Figure 3.66.

Many Linux administrators take full advantage of the GUI. The default Red Hat GUI is GNOME. It's easy to start a command-line interface in GNOME. Right-click any open area of the desktop, and select New Terminal in the menu that appears. This opens the default GNOME terminal command-line interface, shown in Figure 3.67.

Now you're ready to learn all about Linux!

FIGURE 3.66
Graphical login screen

FIGURE 3.67
GNOME with a command-line terminal



Upgrading Red Hat Enterprise Linux

There are two types of upgrades that you may want to consider. First is the obvious type of upgrade, from Red Hat Enterprise Linux 2.1. Second is an upgrade based on the quarterly updated installation CDs available with an official subscription to Red Hat Enterprise Linux 3. Both proceed using the same basic process.

If you've installed Red Hat Enterprise Linux before on the local hard drive, you may just want to upgrade. A good upgrade can save your configuration and data files in their current locations. While you should always back up your data prior to upgrading any operating system, life is a lot less troublesome when you don't have to spend time restoring from a backup.

Normally, Anaconda will detect a previous installation of Red Hat Enterprise Linux on your computer from the `/etc/redhat-release` file. If it doesn't, you can enter the following at the Anaconda `boot:` prompt:

```
boot: linux upgradeany
```

Allowable Upgrades

You can use the Red Hat Enterprise Linux 3 installation CDs to upgrade from Red Hat Enterprise Linux 2.1 on an x86 computer. Red Hat does not support upgrades on other platforms. While Red Hat supports upgrades at this limited level, it recommends that all upgrades be performed as a fresh installation.

Alternatively, if you're installing one of the official quarterly updates, upgrades are one way to update the hundreds of MB of packages that may have been revised. In this case, start with the revised first installation CD that you received from Red Hat (possibly by download).

In either case, the best way to set up an upgrade is with the following command at the first Red Hat Enterprise Linux installation `boot:` prompt:

```
boot: linux upgradeany
```

Making an Upgrade

Upgrades start in the same way as a regular installation. The issues with booting Anaconda from a CD or a floppy don't change. The first few steps of a graphical installation are the same. The first place we diverge from a regular installation is just after configuring a mouse. If Anaconda detects a previous version of Red Hat Enterprise Linux, it will identify it in the Upgrade Examine screen, shown in Figure 3.68.

Click Next to continue. The following screen allows you to update your bootloader from a previous version of GRUB or from LILO. As you can see in Figure 3.69, you can skip the update process or create an entirely new bootloader configuration. Make your selection, and click Next to continue.

FIGURE 3.68
Finding an earlier version of Red Hat

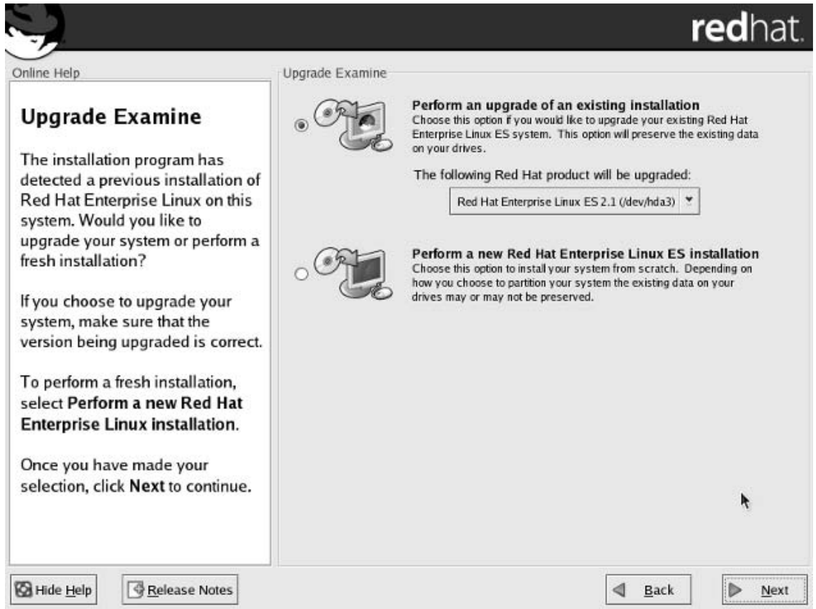
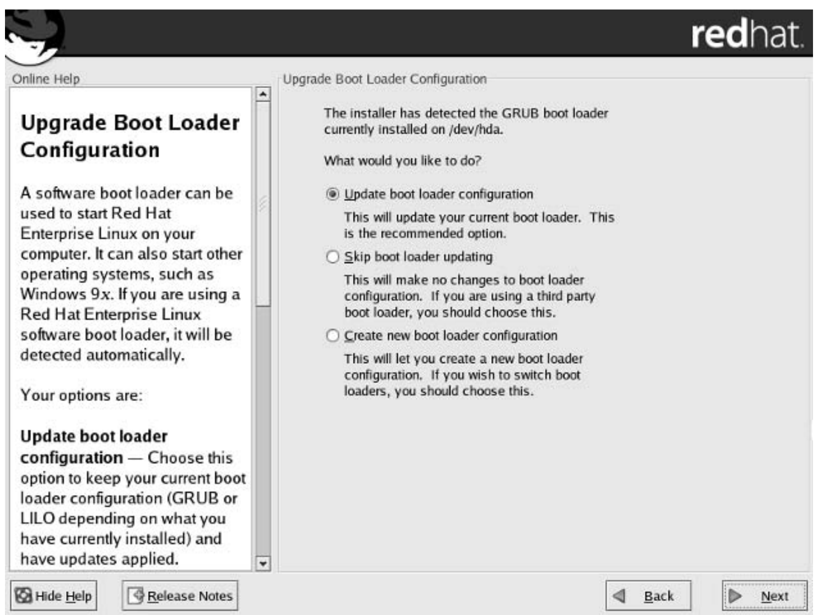
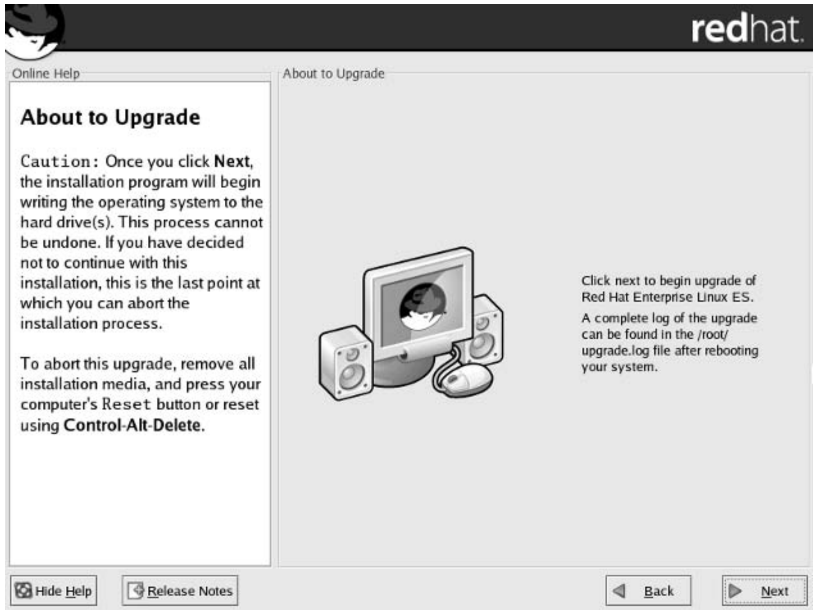


FIGURE 3.69
Updating the bootloader



Now Anaconda takes some time to examine the packages currently on your system. It goes through the list, looking for packages to upgrade. You'll see the About To Upgrade screen, shown in Figure 3.70. When you're ready, click Next to continue and start the upgrade.

FIGURE 3.70
Ready to upgrade



NOTE *You may see a message that upgrades are supported only from Red Hat Linux 6.2 and higher. In reality, Red Hat supports upgrades only from Red Hat Enterprise Linux 2.1.*

The upgrade proceeds as if it were an installation. The process is relatively short; if you've upgraded from a relatively up-to-date system, Anaconda may end up upgrading a small number of packages.

If the upgrade affects any services with configuration files, you should be able to find the original configuration files with an `.rpmsave` extension. For example, if your upgrade affects the Apache web server, you should see the original `httpd.conf` file in the `/etc/httpd/conf` directory saved as `httpd.conf.rpmsave`.

Summary

You have a lot of choices to make when you install Red Hat Enterprise Linux on a computer. Normally, a Red Hat Enterprise Linux installation need not be complex. In most cases, all you need to do is set your computer to boot from the CD drive, insert the first Red Hat Enterprise Linux installation CD, follow some fairly straightforward prompts, and you too can install Red Hat Enterprise Linux in under an hour.

In this chapter, we examined a number of variations of Red Hat Enterprise Linux installations. If you have aspirations of becoming a Linux administrator, we hope this chapter has helped you learn how to handle a variety of situations during the installation process.

This chapter showed you how to create boot and driver disks. We illustrated various ways to test downloaded CDs and examined options available during the installation process. We also showed you how to navigate the Red Hat Setup Agent to help users who select a graphical login screen finish configuring Linux as a desktop.

Many good resources are available for troubleshooting an installation, and you can use these resources while Anaconda is at work. You can take advantage of these resources by viewing the log files in the `/tmp` directory and by accessing installation virtual consoles.

Once you log into Red Hat Enterprise Linux, you'll want to be ready with a command-line interface to learn more with this book and through the Linux community.

Further, Anaconda can help you upgrade from Red Hat Enterprise Linux 2.1. It can also help you install the official quarterly updates provided by Red Hat, and it saves any configuration files you had previously modified.

The next chapter takes a more advanced look at Red Hat Enterprise Linux installations, using text mode, with a focus on installing Red Hat Enterprise Linux over a network.

Chapter 4

Installing Linux over a Network

IF YOU'RE A LINUX system administrator, you may be looking at installing Red Hat Enterprise Linux on multiple computers. You'll want to automate the process. While the installation process shown in Chapter 3 is attractive, it takes a lot of time to install from CDs, especially on multiple computers. You can save time by installing Red Hat Enterprise Linux over a network. You don't have to sit around waiting to insert other Red Hat installation CDs on your computers.

In this chapter, we assume that you've already prepared your computer per the requirements of Chapter 2. For example, if you're planning a dual-boot between Red Hat Enterprise Linux and Microsoft Windows, you've already used the techniques in that chapter to set aside free disk space with sufficient room for Linux.

We'll look at installing Linux from three types of network servers: NFS (Network File System), FTP (File Transfer Protocol), and web (via Apache). We'll learn how to set up the Red Hat Enterprise Linux installation files on each of these servers. You can use the network server as a central source for new packages and programs after Linux is installed. You can even set up these network servers to set up an installation boot server for computers with PXE network cards.

Although you could set up these servers on different operating systems, we'll go through the basics of setting up each service. Future chapters cover detailed configuration of each service.

Also in this chapter, we'll look at the details of the network installation process, from boot disks to a step-by-step analysis of text-mode installation. Why text mode? It's faster—after all, your time is valuable. We'll also examine the subtle differences you'll run into when upgrading an existing Linux installation. Finally, we'll look at methods to help you troubleshoot a network installation.

On the other hand, it's possible to install Red Hat Enterprise Linux in graphical mode over an NFS network connection. If you want to configure Logical Volume Management (LVM) during the installation process, you can only do this in graphical installation mode.

Once you've read Chapters 4 and 5, you'll be ready to install Red Hat Enterprise Linux on several computers simultaneously. This chapter covers the following topics:

- Preparing an NFS server
- Preparing an Apache web server
- Preparing an FTP server

- Configuring a PXE boot server
- Starting a Linux Network Installation
- Troubleshooting a network installation

Preparing an NFS Server

In the following sections, we'll look at configuring an NFS server with the Red Hat installation files from the CDs. When you've configured the server, you'll be able to use the shared NFS directory after Red Hat Enterprise Linux is installed for the RPM packages you may need in the future.

This assumes you already have a Linux or Unix computer, with the appropriate NFS services installed. We'll look at the basic commands that you need to set up an NFS installation server, but the details of how NFS works are not covered in this chapter. If you want to know more about NFS, see Chapter 22.

You can also set up a graphical installation from an NFS server. The chapter also assumes that you're making changes as the root user.

Copying Files

The first step is to set up a directory with the Red Hat Enterprise Linux installation files. You'll need a `/RedHat` directory, with `base` and `RPMS` subdirectories. You need to copy the files in the `/RedHat/base` directory from the first Red Hat installation CD. You'll also need to copy the RPM packages from all three installation CDs to the `/RedHat/RPMS` directory.

This is actually a fairly easy process:

1. Find room for the Red Hat installation files. You'll need nearly 2GB of space (more if you're using the Red Hat update installation CD from the Red Hat Enterprise Linux quarterly update).
2. Create a separate directory. Make sure it's in a partition with sufficient space. For more information on managing partitions, see Chapter 7. For the purpose of this exercise, I've named the directory `/mnt/inst`.

   ```
   # mkdir /mnt/inst
   ```

3. Mount the first Red Hat Enterprise Linux 3 installation CD.

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   ```

NOTE *You can use the original first Red Hat Enterprise Linux installation CD or an Updates CD that you received or downloaded from your Red Hat Network account.*

4. Copy the applicable files from the CD:

   ```
   # cp -ar /mnt/cdrom/RedHat /mnt/inst
   ```

5. Copy the `.discinfo` file from the first installation CD. This allows you to use the Red Hat Package Management tool (`redhat-config-packages`) over the network, which we describe in Chapter 10.

   ```
   # cp /mnt/cdrom/.discinfo /mnt/inst
   ```

6. Unmount the first installation CD. Mount the second Red Hat installation CD. Copy the applicable files from that CD.

   ```
   # umount /mnt/cdrom
   # mount -r /dev/cdrom /mnt/cdrom
   # cp -ar /mnt/cdrom/RedHat /mnt/inst
   ```

7. Repeat step 6 with the third and fourth Red Hat installation CDs.

   ```
   # umount /mnt/cdrom
   # mount -r /dev/cdrom /mnt/cdrom
   # cp -ar /mnt/cdrom/RedHat /mnt/inst
   ```

Now you're ready with a Red Hat Enterprise Linux installation source.

TIP *If you're using a third-party "rebuild" of Red Hat Enterprise Linux 3, you'll have almost all the same software on three installation CDs.*

You could also install Red Hat Enterprise Linux from `.iso` files on a shared NFS directory. I don't include that option in this book, since I believe that it isn't as useful. While you can mount `.iso` files like regular Red Hat installation CDs, that approach doesn't provide a single source for RPM packages after Red Hat Enterprise Linux is installed.

Sharing Directories

If you've installed NFS on your computer, you can now export the shared directory with the Red Hat Enterprise Linux installation files. Exports are documented in the `/etc/exports` configuration file. Open it in the text editor of your choice.

NOTE *Several text editors are available in Linux. For more information on the* `vi` *text editor, see Chapter 6.*

Based on the previous section, we'll share the `/mnt/inst` directory with the Red Hat Enterprise Linux installation files. It's not difficult; just follow these steps:

1. Add the following line to `/etc/exports`:

   ```
   /mnt/inst      *(ro,sync)
   ```

   Make sure there are no spaces after the asterisk; NFS may misinterpret them. Save your changes to `/etc/exports`.

2. Next, export the shared directory with the following command:

```
# exportfs -a
```

3. Now you can make sure that NFS is ready to share your directory. Stop the service. If NFS isn't yet running, the following messages may look like they're creating error messages. Don't worry about it.

```
# service nfs stop
```

4. Copy the applicable files from the CD (this process will probably take several minutes).

```
# service nfs start
```

5. Check your exports. Show the directories that can be mounted with the following command:

```
# showmount -e
```

6. If you've installed a firewall on this computer, it's easiest to disable it. However, that may be risky for your network. You can customize your firewall, as discussed in Chapter 17. For simplicity, I use the following command to "flush" all firewall rules from your Linux computer:

```
# iptables -F
```

   (You can restore your existing firewall on a current Red Hat operating system with the `service iptables restart` command.)

Now you've set up a directory with Red Hat Enterprise Linux installation files, and have shared it using NFS.

NOTE *If you want to continue sharing the NFS installation directory the next time you boot Linux, the* `chkconfig --level 2345 nfs on` *command can help. For more information about* `chkconfig`*, see Chapter 13.*

Setting Installation Parameters

To use the NFS directory you've shared, you'll need two things: the address of the NFS server and the location of the `/RedHat` directory. The address of the NFS server could be a computer name, such as `NFSserver`, or a fully qualified domain name, such as `NFSserver.example.com`. But this requires a working DNS (Domain Name Service) server, which may not apply to all networks.

Alternatively, you can use the IP address of the NFS server. If you don't know that address, run the `ifconfig` command. It should give you output similar to Figure 4.1.

FIGURE 4.1 IP address information

```
[root@Enterprise3 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0D:9D:86:36:A0
          inet addr:192.168.1.23  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:1302 (1.2 Kb)
          Interrupt:10 Base address:0x5000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:599 errors:0 dropped:0 overruns:0 frame:0
          TX packets:599 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:55220 (53.9 Kb)  TX bytes:55220 (53.9 Kb)

[root@Enterprise3 root]#
```

The important piece of information is the IP address; in Figure 4.1, it's 192.168.1.23. For basic information on IP addressing and the other concepts in this section, see Chapter 15. To summarize, once you've set up shared directories on a running NFS server, you need the following bits of information during the installation process:

The IP address of the NFS server If you have a working DNS server for your network, you could substitute the computer name or fully qualified domain name of the server.

The location of the ***/RedHat*** **directory** Based on the actions taken earlier in this chapter, the location of the `/RedHat` directory is **`/mnt/inst`**. If you've set up the Red Hat installation files in a different directory, the location changes accordingly.

You'll get a chance to see how this works in the section "Text Mode: Step by Step," later in this chapter.

Preparing an Apache Web Server

In the following sections, we'll look at configuring an Apache web server with the Red Hat installation files from the CDs. Once you've completed these steps, you'll be able to use a directory on your website after Red Hat Enterprise Linux is installed for the RPM packages you may need in the future.

These sections assume you have a Linux or Unix computer, with the appropriate Apache (`httpd`) services already installed. We'll look at the basic commands that you need to set up an Apache (`httpd`) installation server; however, we don't address the details of how Apache is configured. To learn more about Apache, read Chapter 25.

Once again, these sections assume you're making changes as the root user.

Copying Files

The first step is to set up a directory with the Red Hat Enterprise Linux installation files. You'll need a `/RedHat` directory, with `base` and `RPMS` subdirectories. Copy the files in the `/RedHat/base` directory from the first Red Hat installation CD. Then, copy the RPM packages from all three installation CDs to the `/RedHat/RPMS` directory.

This is actually a fairly easy process.

1. Find room for the Red Hat installation files, preferably associated with the `/var` directory. You'll need nearly 2GB of space (more if you're using the Red Hat update CD from the Red Hat Enterprise Linux quarterly update).

2. Create a separate directory. Make sure it's in a partition with sufficient space. For more information on managing partitions, see Chapter 7. For the purpose of this exercise, I've named the directory `/var/www/html/inst`.

```
# mkdir /var/www/html/inst
```

3. Mount the first Red Hat Enterprise Linux 3 installation CD (this will probably take several minutes).

```
# mount -r /dev/cdrom /mnt/cdrom
```

NOTE *You can use the original first Red Hat Enterprise Linux installation CD or an Updates CD that you received or downloaded from your Red Hat Network account.*

4. Copy the applicable files from the CD:

```
# cp -ar /mnt/cdrom/RedHat /var/www/html/inst
```

5. Copy the `.discinfo` file from the first installation CD. This allows you to use the Red Hat Package Management tool (`redhat-config-packages`) over[stet] the network, which I describe in Chapter 10.

```
# cp /mnt/cdrom/.discinfo /mnt/inst
```

6. Unmount the first installation CD. Mount the second Red Hat installation CD. Copy the applicable files from that CD.

```
# umount /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
# cp -ar /mnt/cdrom/RedHat /var/www/html/inst
```

7. Repeat step 6 with the third and fourth Red Hat installation CDs.

```
# umount /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
# cp -ar /mnt/cdrom/RedHat /var/www/html/inst
```

Now you're ready with a Red Hat Enterprise Linux installation source.

TIP *If you're using a third-party "rebuild" of Red Hat Enterprise Linux 3, you'll have almost all of the same software on three installation CDs.*

Unlike with NFS or a hard disk–based installation, you can't use an Apache server to install Red Hat Enterprise Linux from `.iso` files.

Sharing Directories

If you've installed the Apache web server on your computer, you can now share the associated directory. By default, standard files are stored in `/var/www/html`. Assuming you used the directories cited in the previous section, all you need to cite during the Red Hat Enterprise Linux installation process is the `/inst` directory.

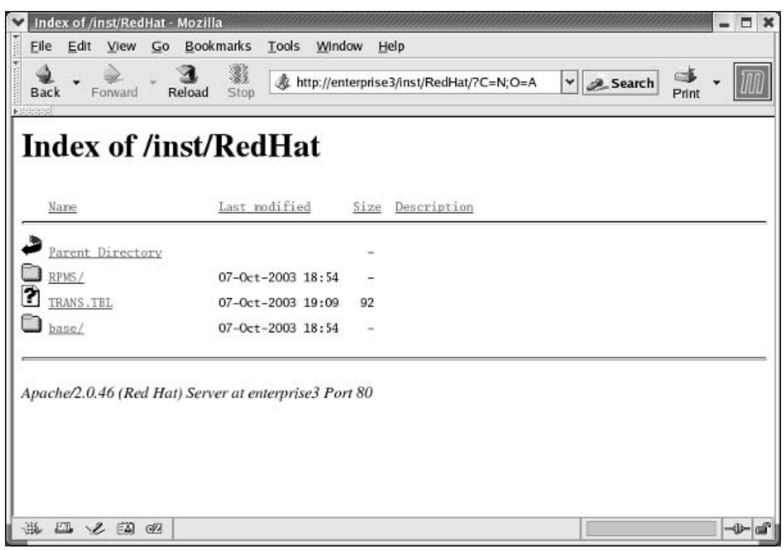
The process is simpler than for NFS. All you need to do is make sure Apache is started with the command

```
# service httpd start
```

and then check to see if you get the "Test Page" when you navigate to `http://localhost` in the web browser of your choice.

Once you've created the share, you'll be able to download individual Red Hat RPM packages via your web server. Figure 4.2 shows how this is possible. Navigate to `http://`*yourwebserver*`/inst/RedHat`, select the `RPMS` directory, and then you can click the RPMs you've loaded in the previous section. You should be able to download the RPMs to your local computer.

FIGURE 4.2
Accessing RPMs through a browser



As with NFS, make sure that a firewall on the local computer isn't blocking access to your web server. The easiest way to do this is to "flush" the current rules in your firewall with the following command:

```
# iptables -F
```

However, that may be risky for your network. You can customize your firewall, as discussed in Chapter 17. You can restore your existing firewall on a current Red Hat operating system with the `service iptables restart` command.

Now you've set up a directory with Red Hat Enterprise Linux installation files, and have shared it using the Apache web server.

NOTE *If you want to continue running Apache the next time you boot Linux, use the* `chkconfig --level 2345 httpd on` *command. See Chapter 13 for more information about* `chkconfig`.

Setting Installation Parameters

To use the Apache directory you've configured, you'll need two things: the address of Apache web server and the location of the `/RedHat` directory. The address of the Apache web server could be a computer name, such as `Webserver`, or a fully qualified domain name, such as `www.example.com`. However, this requires a working DNS server on your LAN, which may not be necessary on a smaller network.

Instead, you can use the IP address of the web server. If you don't know that address, run the `ifconfig` command. Find the IP address information for your computer as described earlier with Figure 4.1.

For more information on IP addressing and the other concepts in this section, see Chapter 15. To summarize, once you've set up shared directories on a running web server, you need the following bits of information during the installation process:

The IP address of the Apache web server If you have a working DNS server on your network, you could substitute the computer name or fully qualified domain name of the server.

The location of the */RedHat* directory Based on the actions taken in the previous section, is the location of the `/RedHat` directory is `/inst`.

You'll get a chance to see how this works in the section "Text Mode: Step by Step."

Preparing an FTP Server

In the following sections, you'll learn how to configure an FTP server with the Red Hat installation files from the CDs. You'll also learn how to connect to the same FTP server after Red Hat Enterprise Linux is installed for the RPM packages you may need

We assume that you already have a Linux or Unix computer, with the appropriate FTP services installed. On Red Hat Enterprise Linux, this includes the `vsftpd-*` RPM package. I don't delve into the details of how FTP servers are configured in this chapter; to learn more about that process, read Chapter 22.

These sections also assume you're making changes as the root user.

Copying Files

The first step is to set up a directory with the Red Hat Enterprise Linux installation files. You'll need a `/RedHat` directory, with `base` and `RPMS` subdirectories. Copy the files in the `/RedHat/base` directory from the first Red Hat installation CD. Then, copy the RPM packages from all three installation CDs to the `/RedHat/RPMS` directory.

This is a fairly easy process.

1. Find room for the Red Hat installation files, preferably associated with the `/var` directory. You'll need a partition with nearly 2GB of space (more if you're using the Red Hat update CD from the Red Hat Enterprise Linux quarterly update).

2. Create a separate directory. Make sure it's in a partition with sufficient space. For more information on managing partitions, see Chapter 7. For the purpose of this exercise, I've named the directory `/var/ftp/pub/inst`.

   ```
   # mkdir /var/ftp/pub/inst
   ```

3. Mount the first Red Hat Enterprise Linux 3 installation CD.

   ```
   # mount /dev/cdrom /mnt/cdrom
   ```

4. Copy the applicable files from the CD.

   ```
   # cp -ar /mnt/cdrom/RedHat /var/ftp/pub/inst
   ```

NOTE *You can use the original first Red Hat Enterprise Linux installation CD or an Updates CD that you received or downloaded from your Red Hat Network account.*

5. Copy the `.discinfo` file from the first installation CD. This allows you to use the Red Hat Package Management tool (`redhat-config-packages`) over the network, which we describe in Chapter 10.

   ```
   # cp /mnt/cdrom/.discinfo /mnt/inst
   ```

6. Unmount the first installation CD. Mount the second Red Hat installation CD. Copy the applicable files from that CD.

   ```
   # umount /mnt/cdrom
   # mount /dev/cdrom /mnt/cdrom
   # cp -ar /mnt/cdrom/RedHat /var/ftp/pub/inst
   ```

7. Repeat step 6 with the third and fourth Red Hat installation CDs.

   ```
   # umount /mnt/cdrom
   # mount /dev/cdrom /mnt/cdrom
   # cp -ar /mnt/cdrom/RedHat /var/ftp/pub/inst
   ```

Now you're ready with a Red Hat Enterprise Linux installation source.

TIP *If you're using a third-party "rebuild" of Red Hat Enterprise Linux 3, you'll have almost all of the same software on three installation CDs.*

Unlike with NFS or a hard disk–based installation, you can't use an FTP server to install Red Hat Enterprise Linux from `.iso` files.

Sharing Directories

If you've installed the FTP server packages on your computer, you can now share the associated directory. By default, standard files are stored in `/var/ftp/pub`. Assuming you used the directories cited in the previous section, all you need to cite during the Red Hat Enterprise Linux installation process is the `/inst` directory.

The process is simpler than for NFS. Just make sure the FTP server is started with the command

```
# service vsftpd start
```

and then check to see if you get the appropriate directories after logging into that FTP server.

NOTE *Prior to Red Hat Linux 9,* `vsftpd` *was an* `xinetd` *service, which you can activate as described in Chapter 18.*

Once you've created the share, you'll be able to download individual Red Hat RPM packages from the FTP server. For more information, see Chapter 10.

As with the other servers, make sure that a firewall on the local computer isn't blocking access to your web server. The easiest way to do this is to "flush" the current rules in your firewall with the following command:

```
# iptables -F
```

However, that may be risky for your network. You can customize your firewall, as discussed in Chapter 17. You can restore your existing firewall on a current Red Hat operating system with the `service iptables restart` command.

Now you've set up a directory with Red Hat Enterprise Linux installation files, and have shared it using an FTP server.

NOTE *If you want to continue running the Red Hat FTP server the next time you boot Linux, use the* `chkconfig --level 2345 vsftpd on` *command. For more information about* `chkconfig`*, see Chapter 13.*

Setting Installation Parameters

To use the FTP directory you've configured, you'll need two things: the address of the FTP server computer and the location of the `/RedHat` directory. The address of the FTP server could be a computer name, such as `FTPserver`, or a fully qualified domain name, such as `www.example.com`. However, this requires a working DNS server, which may not apply to all networks.

Instead, you can use the IP address of the FTP server. If you don't know that address, run the `ifconfig` command. Find the IP address information for your computer as described earlier with Figure 4.1.

For more information on IP addressing and the other concepts in this section, see Chapter 20. To summarize, once you've set up shared directories on a running FTP server, you need the following bits of information during the installation process:

The IP address of the FTP server If you have a working DNS server on your network, you could substitute the computer name or fully qualified domain name of the server.

The location of the */RedHat* directory Based on the actions taken in the previous section, the location of the `/RedHat` directory is `/pub/inst`.

You'll get a chance to see how this works in the section "Text Mode: Step by Step."

Configuring a PXE Boot Server

If you have a network card that conforms to the PXE, you can also set up a boot server for those computers. First, you need a network installation source configured as an NFS, HTTP, or FTP server. You've seen how that's done in the first sections of this chapter. The key to this is the Network Installation And Diskless Environment tool, shown in Figure 4.3. This is based on the `redhat-config-netboot` RPM.

FIGURE 4.3
Network Installation And Diskless Environment tool

Preparing a PXE Boot Server

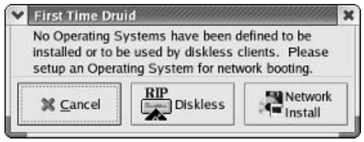
To prepare a PXE boot server, you'll need to add the PXE boot files from the first Red Hat Enterprise Linux installation CD to the network installation directory. These files are located on that CD, in the `images/pxeboot` directory. For example, if you have a CD mounted on `/mnt/cdrom` and have configured the NFS installation server described earlier, you could copy the needed files with the following command:

```
# cp -ar /mnt/cdrom/images/pxeboot /mnt/inst
```

Using the First Time Druid

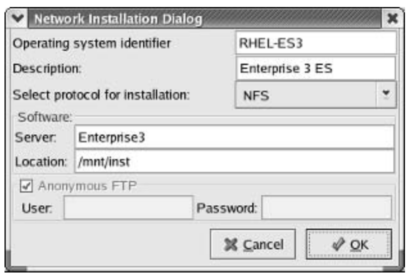
To configure a PXE boot environment on a Red Hat Enterprise Linux 3 server, you can use the Network Installation and Diskless Environment tool. To start it in the GUI, click Main Menu ➢ System Settings ➢ Server Settings ➢ Network Booting Service, or run the `redhat-config-netboot` command. The first time you start this tool, it opens the First Time Druid window, shown in Figure 4.4.

FIGURE 4.4 First Time Druid window



Click the Network Install button. This opens the Network Installation Dialog screen, shown in Figure 4.5.

FIGURE 4.5 Network Installation Dialog screen



If you've used this tool before, you're taken directly to the Network Installation And Diskless Environment tool. You can start the First Time Druid with the File ➢ First Time Druid command.

Copying to the TFTP Server

Before you begin, you may need to install the TFTP server on your computer. On Red Hat Enterprise Linux 3, it's part of the Network Servers package group that isn't installed by default. Using the techniques described in Chapter 10, make sure the `tftp-server` RPM is installed.

The Network Installation Dialog window from Figure 4.5 is used to transfer boot files and a Linux kernel to the `/tftpboot/linux-install` directory. To set this up, you'll want to use the guidance in Table 4.1.

TABLE 4.1: NETWORK INSTALLATION DIALOG SETUP

OPTION	DESCRIPTION
`Operating System Identifier`	Enter a descriptive directory name; this must be one word. This becomes a subdirectory name in the `/tftpboot/linux-install` directory.

Continued on next page

Table 4.1: Network Installation Dialog Setup *(continued)*

Option	**Description**
`Description`	Add a short description of your choice.
`Select Protocol For Installation`	Select the installation server type you've configured: NFS, HTTP, or FTP.
`Server`	Enter the name or IP address of the local installation server.
`Location`	Include the directory with your network installation server files.
`Anonymous FTP`	If deselected, you can enter an authorized username and password for a non-anonymous FTP installation server.

When you're ready, click OK. This tool now transfers the kernel and boot files from your installation server to the `/tftpboot/linux-install` directory.

If you get an error message, then your installation server service (NFS, HTTP, or FTP) may not be active, you may not be working from local source files, or you may have forgotten to transfer files from the `images/pxeboot` directory on the first Red Hat Enterprise Linux installation CD.

If you prefer to work from the command-line interface, you can configure your system using the parameters shown in Figure 4.5 with the following command:

```
# pxeos -a -i "Enterprise 3 ES" -p NFS -D 0 -s Enterprise3 -L /mnt/inst RHEL3-ES
```

This command installs several standard message files in the `/tftpboot/linux-install/pxelinux.cfg` directory. Perhaps the key is a file named `default`, which (as you'll soon see) includes the menu you see when you install from a computer with a PXE network card. When I set up my TFTP server, I found the following error in this file:

```
label 1
   kernel "RHEL3-ES"/vmlinuz
   append initrd="RHEL3-ES"/initrd.img ramdisk_size=10000
```

The error is in quotes; if you see this in your version of your file, remove the quotes in the text editor of your choice.

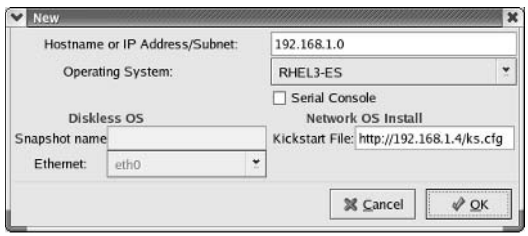
Adding Hosts

Once you've configured the TFTP server, the next step is to add hosts, specifically those associated with the computers with PXE network cards. Back in the Network Installation And Diskless Environment tool, click New. This opens the New window, shown in Figure 4.6. To set this up, you'll want to use the guidance in Table 4.2.

Table 4.2: Network Installation Dialog's New Setup

Option	Description
Hostname Or IP Address/Subnet	Enter the fully qualified domain name of the computer or network IP address of the subnet.
Operating System	Use the same name as the Operating System Identifier you used in the Network Installation Dialog window.
Serial Console	Activate if you're installing and want the display sent through a serial port connection; this is useful for servers on racks.
Kickstart File	Add the path to the Kickstart file that you want to use for your PXE computers.

Figure 4.6
Configuring a PXE host



Starting the Boot Server

Now you'll want to make sure the appropriate servers are started. Earlier in this chapter, I showed you how to start an NFS, an FTP, and an HTTP installation server. You'll also want to start the TFTP boot server. On Red Hat Enterprise Linux 3, it's an `xinetd` server, which is described in Chapter 18; first make sure it's active with the following command:

```
# chkconfig tftp on
```

You may also want to make sure it's activated the next time you boot, along with other `xinetd` services, with the following command:

```
# chkconfig --level xinetd 345 on
```

Configuring DHCP

Finally, you need to configure your DHCP server to accept requests from PXE clients. Configuring a DHCP server is a straightforward process, which we cover in Chapter 19. This means you'll need to configure a DHCP server on a Linux computer; a DHCP server on a home router may not do the job. You'll want to add the following lines to your `/etc/dhcpd.conf` configuration file:

```
allow booting;
allow bootp;
class "pxeclients" {
```

```
    match if substring(option vendor-class-identifier, 0, 9)="PXEClient";
    next-server 192.168.1.4
    filename "linux-install/pxelinux.0";
)
```

In this case, 192.168.1.4 is the IP address of my installation server.

Finally, you'll want to make sure the DHCP server is running. If this is the first time you've set up a Linux DHCP server, you'll want to test it from a remote client with the `dhclient` command.

Starting a PXE Network Installation

Once you boot your computer, the process for starting a PXE network installation will vary. Generally, this involves a boot command such as F12, which you may need to activate in your computer's BIOS.

When the boot process starts, your PXE network card looks for the DHCP server you configured. If it finds your DHCP server, it'll take the information as defined in `/etc/dhcpd.conf`. You should then see a boot menu similar to Figure 4.7.

FIGURE 4.7 Installing from a PXE host

You can specify the Kickstart configuration file of your choice from the boot prompt. For example, the following command calls the specified Kickstart file from a web server on a remote computer:

```
boot: 1 ks=http://192.168.1.4/ks.cfg
```

Starting a Linux Network Installation

When you're installing Red Hat Enterprise Linux over a network connection, you probably want to install this system on several computers simultaneously. In this case, you generally aren't going to use the Red Hat installation CDs. Therefore, you'll need a boot disk. You can customize automated

installations through a Kickstart file stored on that boot disk. We describe this process in Chapter 5. Red Hat provides boot disk images that you can write to floppies and small CDs.

Once you have the boot disk, you can start the Red Hat Enterprise Linux network installation process. In this chapter, we proceed with text-mode installation, since we covered graphical mode in Chapter 3. In any case, graphical-mode installations aren't allowed if you're installing from an FTP or Apache server.

The Red Hat Enterprise Linux installation program is known as Anaconda. You can customize the Anaconda installation process to omit installation options such as games. For more details, read Chapter 5.

Making Boot Disks

Red Hat provides boot disk images on the first installation CD, in the `/images` directory. You can even use the first installation CD itself as a network boot disk. There are two basic files that you can use to create an installation boot disk; several driver images are also available that you can use to create 1.44MB driver floppy disks.

Be sure to be consistent; if you set up an installation server with the first Red Hat Enterprise Linux update CD, make sure to create a boot disk from the files on that CD. The appropriate files in the `/images` directory are briefly described in Table 4.3.

TABLE 4.3: BOOT IMAGES

FILENAME	DESCRIPTION
`bootdisk.img`	Used to create a standard boot floppy for local and network installations.
`drvblock.img`	Contains additional block device drivers; may be needed for many SCSI hard drives.
`drvnet.img`	Includes additional network device drivers.
`pcmciadd.img`	Adds additional PCMCIA drivers for many laptop computers.
`boot.iso`	Includes data from all boot and driver disks. Since it's too big for a 1.44MB floppy, it's set up to be recorded on a CD.

You can write an `.img` file to a 1.44MB floppy disk in one of three basic ways. If you have a Linux computer, you can use the `cat` or `dd` command. For example, you can use either of the following commands to write the contents of `bootdisk.img` to a 1.44MB floppy drive. These commands assume you've mounted the first Red Hat Enterprise Linux installation CD on the `/mnt/cdrom` directory.

```
# dd if=/mnt/cdrom/images/bootdisk.img of=/dev/fd0
# cat /mnt/cdrom/images/bootdisk.img > /dev/fd0
```

A second approach is to write the contents of these image files to a 1.44MB floppy drive in Microsoft Windows. The key utility is on the first Red Hat Enterprise Linux installation CD, in the `/dosutils` directory. The command-line version of this interface is `RAWRITE.EXE`. In Microsoft

Windows, open an MS-DOS command-line window. Insert your first Red Hat Enterprise Linux installation CD. If your CD is on the E: drive, run the following commands:

```
E:\>DOSUTILS\RAWRITE
Enter disk image source file name: E:\IMAGES\BOOTDISK.IMG
Enter target diskette drive: A:
Please insert a formatted diskette in drive A: and press -ENTER- :
```

You can also create a boot CD from the `boot.iso` file. For more information, refer to the `cdrecord` command described in Chapter 14. You can even start a network installation using the first Red Hat Enterprise Linux 3 installation CD; just remember to start the installation using the `linux askmethod` or `text askmethod` command.

Text Mode: Booting

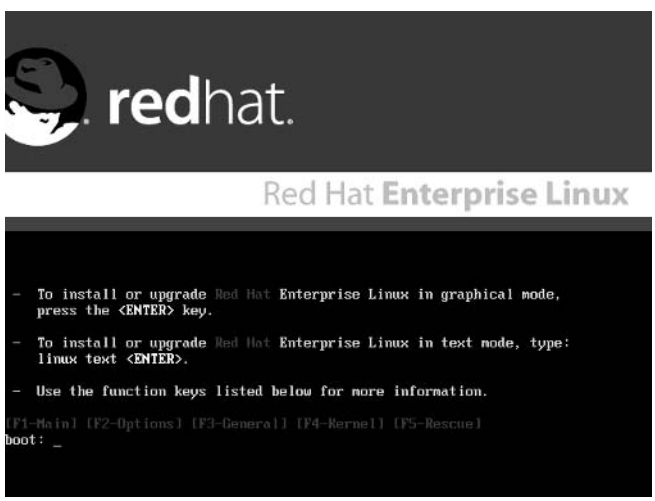
Now we'll examine a text-mode network installation. As described in the last section, there are three basic options for boot disks:

- A floppy disk written from the `bootdisk.img` file
- A CD written from the `boot.iso` file
- The first Red Hat Enterprise Linux installation CD

In all of these cases, you'll see the menu shown in Figure 4.8.

We installed Red Hat Enterprise Linux in graphical mode in Chapter 3. That chapter was more focused on regular users. Now we'll look at the administrative side of things in more detail. As you can see from the initial menu, several other menus are available. You can get to the Installer Boot Options menu by pressing F2. The menu is shown in Figure 4.9.

FIGURE 4.8
Red Hat Enterprise Linux installation menu



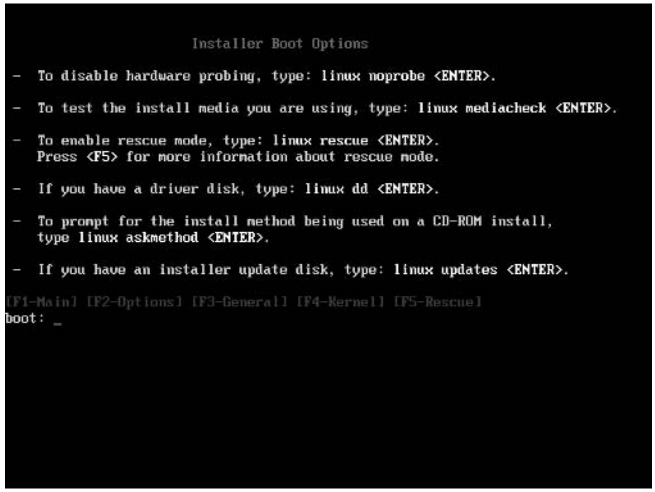


FIGURE 4.9 Installer Boot Options menu

This menu lists some of the available options for what you can enter at the `boot:` prompt. Table 4.4 describes these options.

TABLE 4.4: INSTALLER BOOT OPTIONS

OPTION	DESCRIPTION
`linux noprobe`	Disables detection of key hardware components; a viable option if you're having trouble with hardware detection and have the driver disks you need.
`linux mediacheck`	Adds an additional step to the process, where the integrity of media such as installation CDs are tested against embedded MD5 checksums. If you're installing from downloaded ISO files, this is done automatically.
`linux rescue`	Starts a process that detects current Linux partitions on your system; can be used to recover from a number of different boot failures (for details, see Chapter 11).
`linux dd`	Adds an additional step to the process, prompting for a driver disk. This is appropriate when you have hardware with drivers that aren't already included on the Red Hat disks.
`linux askmethod`	If you're starting the installation process from the first Red Hat installation CD, this allows you to select a network installation source.
`linux updates`	Allows you to use an update floppy disk, mostly for upgrades. If you're using a quarterly update, you'll want to start with the `linux upgradeany` command.
`linux lowres`	Starts a graphical-mode installation process in a screen with 640×480 resolution.

NOTE *If you want to start the installation in text mode, just substitute* `text` *for* `linux` *in one of the options in Table 4.4. Alternatively, you can start the installation in text mode by using the* `linux text` *or* `text` *command. You can even combine commands; for example, the* `linux rescue askmethod` *command can connect you to a network installation source directory when using the first installation CD to rescue your system.*

In other words, if you're starting the text-mode installation process from the `bootdisk.img` floppy or the `boot.iso` CD, you'll want to enter the following:

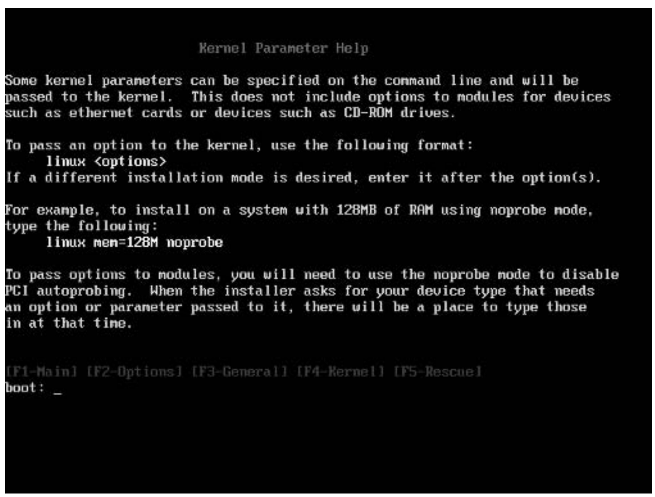
```
boot: text
```

Alternatively, if you're starting the installation process from the first installation CD, enter this:

```
boot: text askmethod
```

One of the things you can do at the boot prompt is to specify the parameters of some hardware. Figure 4.10, the Kernel Parameter Help screen, provides the basics of what you can do. For more information, run the `man bootparam` command on another Linux computer.

FIGURE 4.10
Kernel Parameter Help screen



TIP *If your keyboard doesn't seem to work after this step, try disabling legacy USB support in the computer BIOS, or tap a key periodically while the installation program loads. This can help with some HP/Compaq systems.*

Text Mode: Step by Step

Now that you've started the installation process, let's examine how this works, step by step. We'll describe the text-mode process in detail, based on starting from a `bootdisk.img` installation floppy. The other startup methods are less complex.

TIP *This is a very long section. If you're planning to read it all at once, it may help to take a break before you begin.*

1. Boot your computer with the Red Hat Enterprise Linux installation floppy or CD, created from the `bootdisk.img` or `boot.iso` files, using the techniques described earlier in the "Making Boot Disks" section. Alternatively, you can use the first Red Hat Enterprise Linux installation CD.

TIP *If you have a paid version of Red Hat Enterprise Linux and want to install from CDs, use the update CD. You'll need several hundred MB of free space available to support the upgrade.*

2. When you see the prompt, enter the following:

```
boot: text
```

NOTE *If you're using the first Red Hat Installation CD, enter* **`text`** `askmethod` *at the* `boot:` *prompt.*

NOTE *If you're installing from an NFS server, you can also set up a graphical installation with the* `linux askmethod` *command.*

You'll see a series of messages installing a basic kernel and the text-mode version of the Anaconda installation program.When you see the Welcome To Red Hat Enterprise Linux screen, click OK to start the process.

3. Select a language from the Language Selection screen, shown in Figure 4.11. While English is the default, you can install Red Hat Enterprise Linux with prompts in some 19 different languages and dialects. You can use the Up and Down arrow keys to make your selection. When you've selected your language, use the Tab button to highlight OK, and then press Enter or F12 to continue.

NOTE *With the various text-mode menus, you can use the arrow and Tab keys to navigate between selections. Once you've made your selection, you can press F12, or highlight OK and press Enter or the spacebar to continue. If there are settings you can toggle, highlight the desired setting and press your spacebar.*

4. Select a keyboard from the Keyboard Type screen, shown in Figure 4.12. While the `us` keyboard is the default, you can set up a Red Hat Enterprise Linux installation on nearly 50 different keyboards, many of them customized for other languages. Once you've selected your keyboard, press F12 to continue.

5. Select from the network installation methods, shown in Figure 4.13. As described earlier, you can install from an NFS, an FTP, or an HTTP (Apache) server. Make your selection, and press F12 to continue.

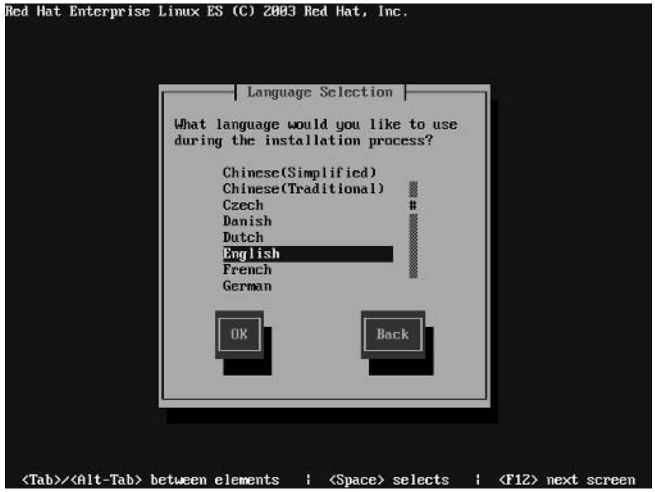
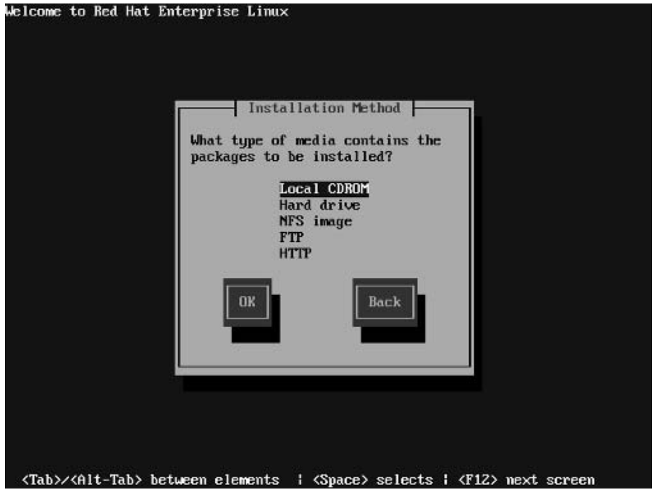


FIGURE 4.11
Choosing a language

FIGURE 4.12
Choosing a keyboard type

NOTE *If you have more than one network card, you'll see a menu where you're allowed to select between your cards, such as* `eth0` *and* `eth1`*. There's no easy way to tell which card is connected to which network. If you select the wrong card, you'll see an error message in the next few steps. Some trial and error may be required; if you select the wrong network card, use the Back options in the menus, and then try the other card.*

FIGURE 4.13
Specifying an installation method



6. Additional drivers may be required with the `bootdisk.img`-based installation floppy, as shown in Figure 4.14. Select Use A Driver Disk, and press Enter to continue. If you don't see the No Driver Found screen, you probably started from the first Red Hat installation CD or the `boot.iso`-based CD; in that case, skip ahead to step 10.

7. Now that you need a new driver, you can select a driver disk, as shown in Figure 4.15. You can use one of the driver floppies you may have created earlier in the section "Making Boot Disks." In that case, select `fd0`. If you can use the first Red Hat installation CD as a driver disk, select the device associated with your CD, usually `hdb` or `hdc`. After you make your selection, press F12 to continue.

FIGURE 4.14
No Driver Found screen

8. At the prompt shown in Figure 4.16, insert the driver disk (floppy or CD) appropriate for your network card or hard drive. If you need both, you'll be prompted to repeat the process with the other disk.
9. If you need a different driver disk, you'll see the Error screen, shown in Figure 4.17. Select Load Another Disk, and press Enter to continue. If you continue to see the same message, your driver disk may be corrupt or may not support your hardware. Return to step 7. If you don't see the Error screen, proceed to step 10.

FIGURE 4.15 Selecting a driver disk source

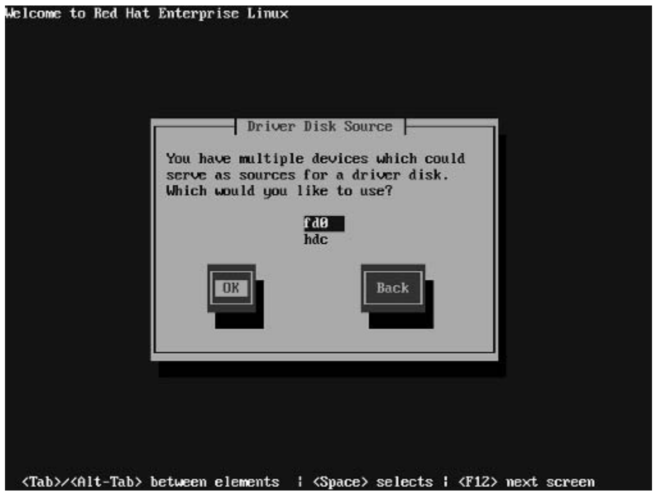


FIGURE 4.16 Inserting a driver disk

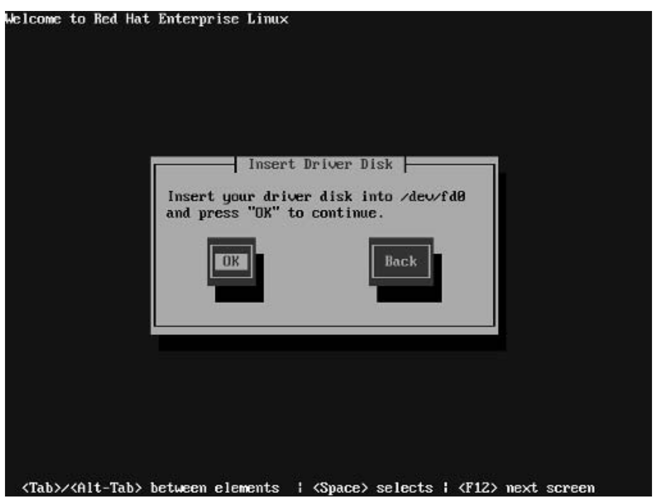


FIGURE 4.17 Prompt for another driver disk

10. Now that you've installed drivers for your network card and/or hard drive, you're ready to set up the connection to the network server. In the Configure TCP/IP screen shown in Figure 4.18, you'll need to set up TCP/IP settings for your computer.

    If a DHCP server is available for your network, you can keep the Use Dynamic IP Configuration (BOOTP/DHCP) setting active; otherwise, you'll need to set up key IP address information for your system. In that case, highlight the setting and press the spacebar to deselect it. For more information on IP addressing, see Chapter 15. Make your selections, and press F12 to continue. If you have a second network card, you'll repeat this step with that card.

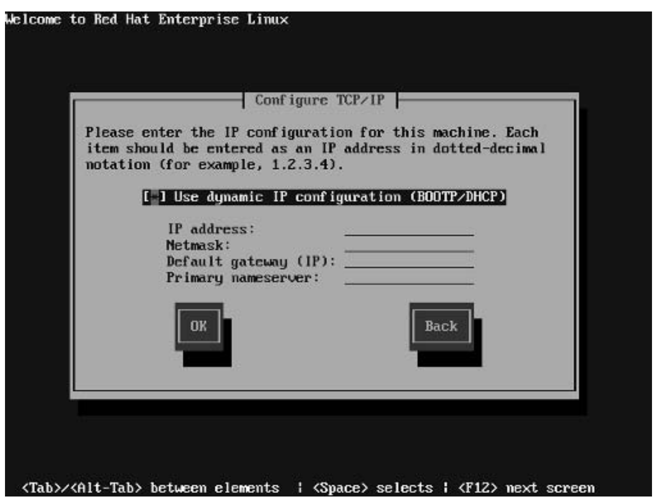


FIGURE 4.18 Configuring TCP/IP

TIP *If you have a home network, you may already have a DHCP server. Many hardware routers, such as those associated with cable modems or DSL adapters, include their own DHCP server. If you have one of these components, check the documentation associated with that router. Make sure the assigned IP addresses matches the subnet you've configured for your network.*

11. What you'll do next varies slightly depending on whether you're installing from an NFS, an Apache HTTP, or an FTP server. In any of these cases, you'll need to cite the name or IP address of the server, as well as the location of the installation files. Figures 4.19, 4.20, and 4.21 illustrate what you may enter for each of these types of servers, based on the server setup instructions earlier in this chapter. For example, Figure 4.19 works if you've installed the `RedHat` directory as part of the `/mnt/inst` directory on the NFS server. Figures 4.20 and 4.21 are based on the settings described earlier for copying installation files to an HTTP or an FTP server.

FIGURE 4.19
NFS Setup screen

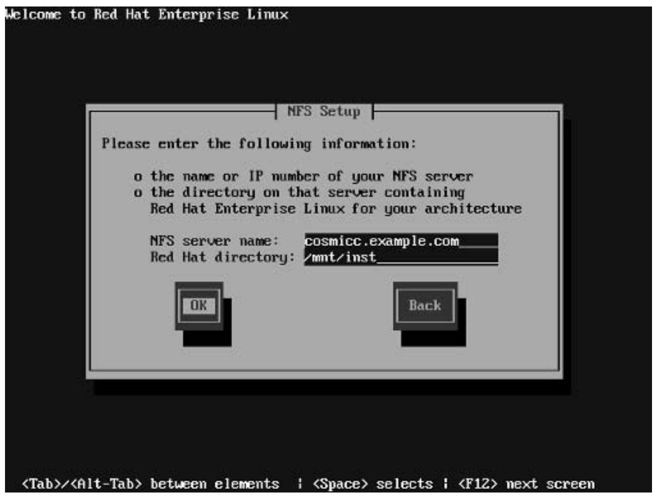
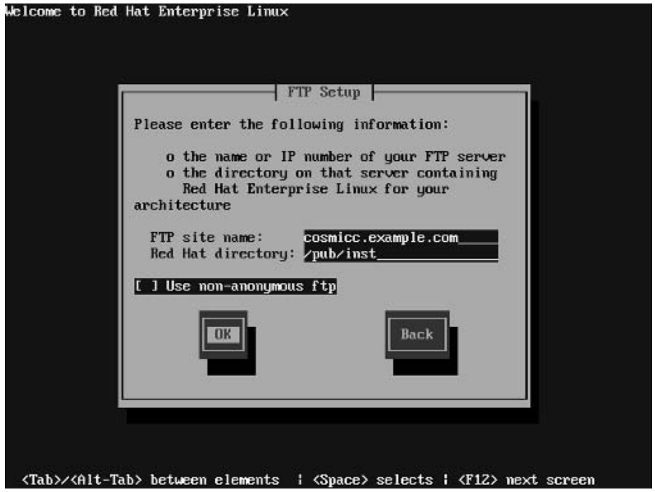


FIGURE 4.20
HTTP Setup screen

FIGURE 4.21
FTP Setup screen



If you're installing from an FTP server, you'll see in Figure 4.21 that you can install from a nonanonymous FTP server—that is, where you have a user account with a password. If you select this option, you'll be prompted for your username and password.

12. If you're successful connecting to any network server, you'll eventually see the following message, briefly:

```
Running anaconda, the Red Hat Enterprise Linux System installer - please wait
```

followed by the welcome screen, shown in Figure 4.22. The steps that follow are independent of the installation server you're using. Select OK to continue.

NOTE *If you started with the* `linux askmethod`*command and are installing from an NFS installation server, Anaconda now starts the graphical installation process, most of which is shown in Chapter 3..*

13. Select a mouse from the Mouse Selection screen, shown in Figure 4.23. You can select from 30 types of pointing devices, including several that connect to USB interfaces. In most cases, if you select a two-button mouse, the Emulate 3 Buttons option is automatically selected.

NOTE *The Emulate 3 Buttons option lets you simulate the functionality of a middle mouse button by pressing the left and right buttons simultaneously. You can select or deselect this option by highlighting it and pressing the spacebar.*

14. If you're installing Red Hat Enterprise Linux on a computer that already has a previous version of Red Hat, you'll see the System To Upgrade screen shown in Figure 4.24. If you're planning to upgrade an existing installation of Red Hat Enterprise Linux, select it. Or, you can set up a fresh installation of Red Hat Enterprise Linux in the same space by selecting Reinstall System.

FIGURE 4.22
The welcome screen

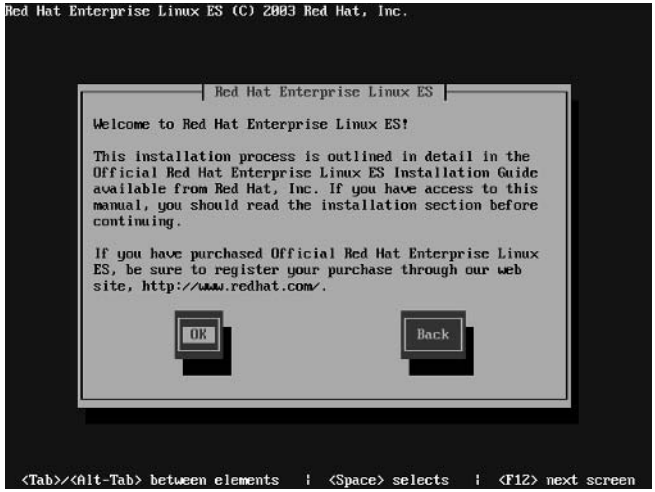


FIGURE 4.23
Selecting a mouse

If you choose to upgrade an existing Red Hat Enterprise Linux installation, read the next section. Otherwise, select Reinstall System, and press F12 to continue.

NOTE *While Red Hat supports upgrades from Red Hat Enterprise Linux 2.1, Red Hat recommends that all configurations work from a fresh installation.*

FIGURE 4.24
Choosing to upgrade

15. Next, you'll select the Disk Partitioning Setup, shown in Figure 4.25. If you select Autopartition, Red Hat Enterprise Linux automatically configures partitions for you, based on the required packages, your RAM, and the size of available partitions on your hard drive(s). If you select Disk Druid, skip to step 17. Make your selection, and press Enter to continue.

NOTE *If this is a new hard disk, you may see a warning that the partition table is unreadable. You'll be given an opportunity to initialize the drive. You'll have to answer "yes" to install Red Hat Enterprise Linux on this hard drive.*

FIGURE 4.25
Disk Partitioning Setup screen

16. If you've chosen to let Red Hat autopartition your system, you'll see the Automatic Partitioning window, shown in Figure 4.26. If you're installing Red Hat Enterprise Linux on a computer with Linux and Microsoft Windows partitions, be careful. The default option for a Server installation would delete your Microsoft Windows operating system. Table 4.5 describes the options. Make your selection, and press F12 to continue.

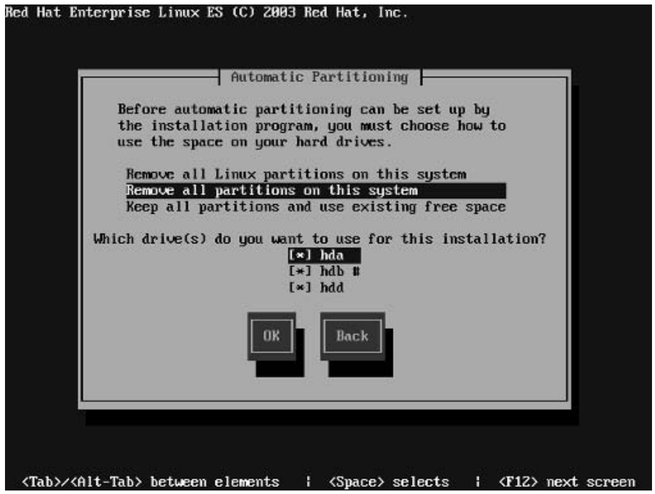


FIGURE 4.26 Automatic Partitioning screen

TABLE 4.5: AUTOMATIC PARTITIONING OPTIONS

OPTION	DESCRIPTION
Remove All Linux Partitions On This System	Deletes all partitions formatted to Linux filesystems; doesn't affect partitions formatted to other filesystems, such as those associated with Microsoft Windows.
Remove All Partitions On This System	Deletes all partitions on this computer.
Keep All Partitions And Use Existing Free Space	Assumes you have unpartitioned free space on your hard drive(s); if you don't, this option leads to an error message.
Which Drive(s) Do You Want To Use For This Installation?	If you have more than one physical hard drive, you're allowed to select the drives where Red Hat Enterprise Linux is to be installed.

17. You're asked to confirm your selection, as shown in Figure 4.27. If you're satisfied with your choice, select Yes and press Enter to continue.

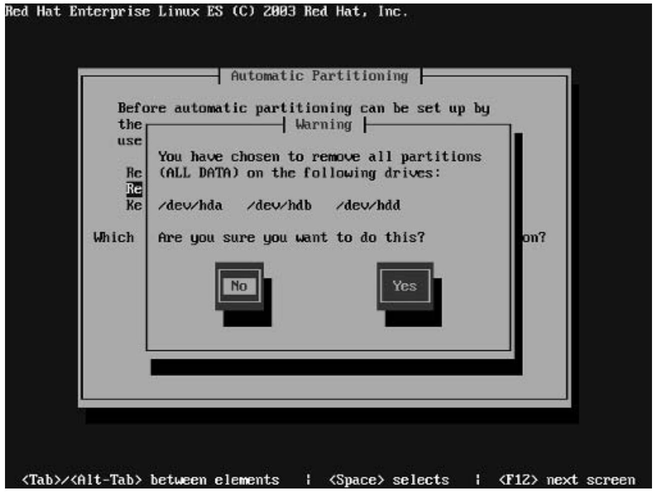


FIGURE 4.27 You'll see this warning before you delete partitions.

18. You're taken to a Disk Druid screen, where you can review the choices made by Anaconda's Automatic Partitioning. One example is shown in Figure 4.28. When space permits, Red Hat normally assigns twice the amount of RAM as a swap partition.

19. You can edit configured partitions. For example, if you had the configuration shown in Figure 4.28, you could highlight `hda2`, associated with the root (/) directory. You could then use the Tab key to highlight the Edit option and press Enter; this would take you to the Add Partition window, where you can change the settings associated with `hda2`. Figure 4.29 shows this menu, and Table 4.6 describes the options.

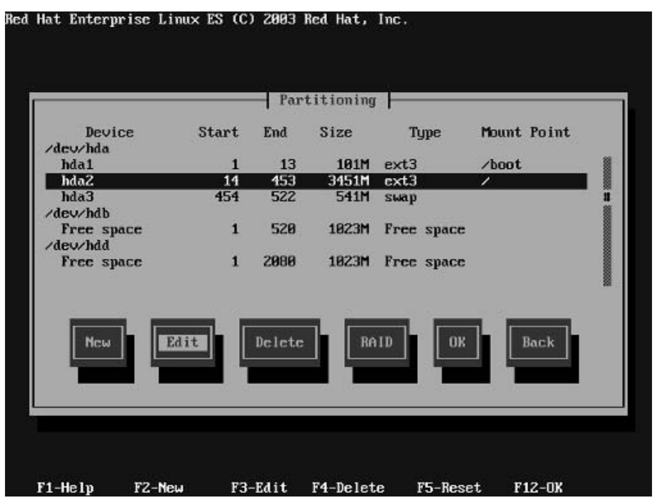


FIGURE 4.28 Disk Druid

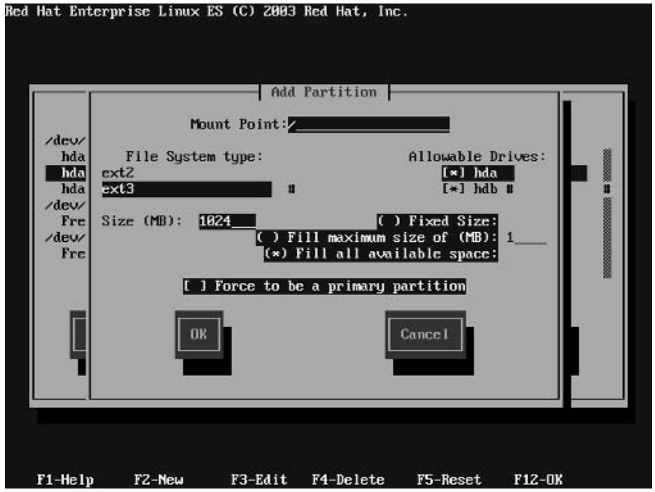


FIGURE 4.29 Editing a configured partition

TABLE 4.6: OPTIONS IN THE ADD PARTITION WINDOW

OPTION	**DESCRIPTION**
Mount Point	Specifies the directory to be mounted on the partition; for mountable directories, consult the discussion on the Filesystem Hierarchy Standard in Chapter 7.
File System Type	Sets the format for the partition; you're allowed to select from the Linux ext2 or ext3 standards, the Linux swap format, the Logical Volume Manager (LVM) volume format, the software RAID format, or the Microsoft Windows–style vfat format.
Allowable Drives	Notes the hard drive device associated with the partition.
Size (MB)	Specifies the size of the partition, in MB; this can be fixed, growable to a specific size, or can be set to fill any remaining free space on the hard drive.
Force To Be A Primary Partition	Generally, you'll want the partition with the `/boot` directory to be on a primary partition below cylinder 1024.

20. You can also add new partitions. For example, I've added a partition for the `/var` directory to help control the remaining free space, as shown in Figure 4.30.

21. If you've configured two or more software RAID partitions, you can configure a RAID device. Back in the main Disk Druid screen shown in Figure 4.28, select RAID and press Enter. If you have sufficient available software RAID partitions, you'll see the Make RAID Device menu, shown in Figure 4.31.

FIGURE 4.30
Adding a partition

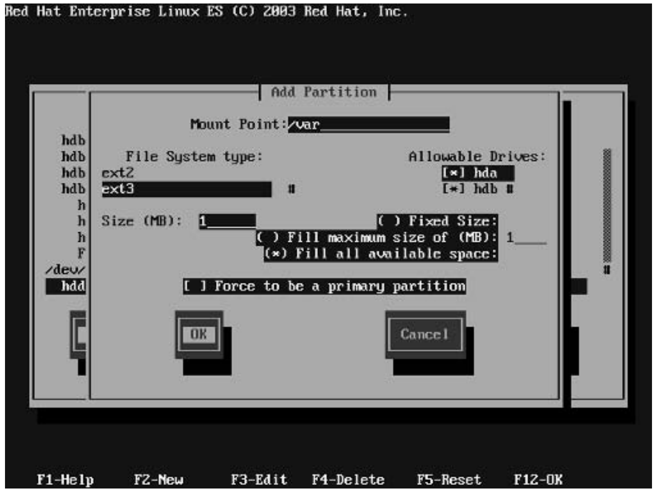
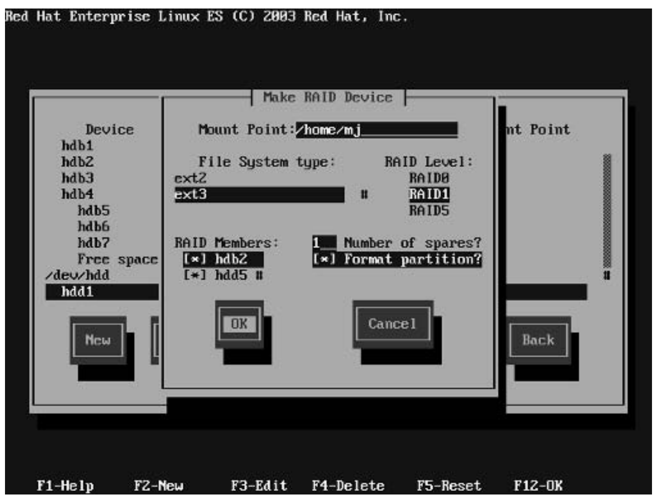


FIGURE 4.31
The Make RAID Device menu



In this case, I've created a RAID1-level device, with three RAID member partitions: `hda2`, `hdb2`, and `hdd5`. Since RAID1 requires only two partitions, one is used as a spare. For more information on RAID, see Chapter 14.

NOTE *Although you can format Logical Volume Manager (LVM) partitions, the Red Hat Enterprise Linux text-mode menus don't allow you to manually configure Logical Volume Manager filesystems. However, it's possible in the graphical-mode installation shown in Chapter 3 as well as Kickstart installations shown in Chapter 5.*

Partition Sizes

One important decision during Red Hat Enterprise Linux installation is the size and number of your partitions. As described in Chapters 3 and 7, you can mount different directories on physically separate hard drive partitions. This can protect you, to keep an overload on a specific partition from crashing your computer.

If you have less than, say, 3GB available on your hard drive, the choices are simple. You don't have much extra room on your hard drive and are therefore pretty much limited to separate partitions for a root (/) and /boot directory and a swap partition. Chapter 7 describes this and several typical partition configurations.

If you have more space, you have more flexibility. You may be able to configure separate partitions for several different directories. As described in Chapter 7, some directories shouldn't be mounted on separate partitions.

The following are some examples of directories you may want to mount and include the *minimum* space required when you install everything in Red Hat Enterprise Linux (this may be larger than the size of the files installed in the particular directory).

If you want to install everything, you need room for more than just the 4000MB of files associated with an "Everything" installation. You need room at least for a swap partition. When I tried an "Everything" installation, I needed 5100MB of space (including for a swap partition).

Remember, you'll also need additional room for users, applications, quarterly and `up2date` updates, and log files. For example, large websites may produce gigabytes of log files every day!

Directory	Description
`/boot`	Contains the boot files, including the Linux kernel; the default 100MB size should be sufficient.
`/`	The top-level root (/) directory; includes all directories not mounted on separate partitions. When other directories shown are mounted on separate partitions, the remaining directories under root (/) contain about 500MB of files under the "Everything" installation.
`/home`	Includes home directories for all users except root; when selecting a size, you need to consider longer-term needs of current and future users.
`/home/mj`	Limits the amount of space available for a specific user; you can also do this with quotas described in Chapter 9.
`/opt`	Designed for files with many third-party applications; Red Hat Enterprise Linux leaves this empty for this purpose.
`/tmp`	Includes files that are automatically deleted on a regular basis; suitable for downloads. Almost empty after installation.
`/usr`	Most of Red Hat Enterprise Linux is installed here; an "Everything" installation requires about 3.4GB of space. Some third-party programs may also require space in this directory.
`/var`	Includes directories for log files and print spools; should leave several hundred MB of empty space in this directory. About 200MB of files are installed via the "Everything" installation. If you're experienced with Linux, you should know that this directory fills up quickly, with log files as well as any files you might install in systems such as Web and FTP servers.

22. Create and delete additional partitions as desired. I've created the series of partitions shown in Figure 4.32. When you've finished, press F12 to continue.

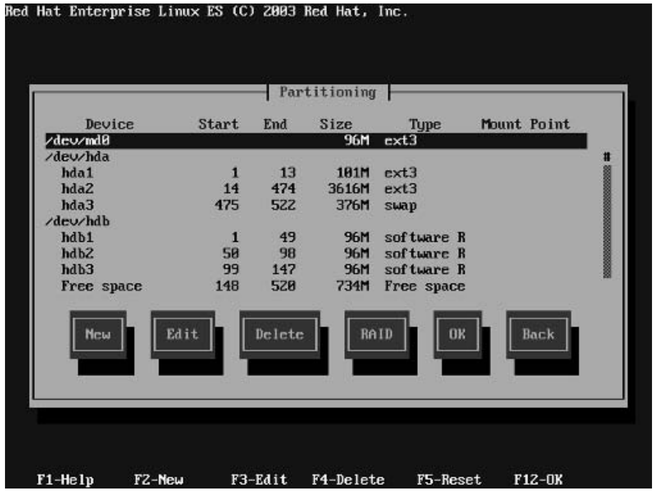


FIGURE 4.32
A partition configuration

23. Now you can select your bootloader. You have three choices, as shown in Figure 4.33. The default is GRUB, the Grand Unified Boot Loader, which is the default described in Chapter 11. The main alternative is LILO, the Linux Loader, which was the default on older versions of Red Hat Enterprise Linux. You can choose to use another bootloader, such as those associated with the proprietary Partition Magic or System Commander programs. Since Red Hat has deprecated LILO, we will stick with the default GRUB bootloader. Make your selection and press F12 to continue.

NOTE *The terms* boot loader *and* bootloader *are used interchangeably.*

24. If there are special parameters associated with your kernel, you can enter them in the second Boot Loader Configuration screen, shown in Figure 4.34. You can enter kernel parameters described earlier in the section "Text Mode: Booting." Red Hat Enterprise Linux may do this automatically for you. For example, you can disable the older Advanced Power Management scheme by adding the `apm=off` command.

NOTE *Generally, you won't need to activate the Force Use Of LBA32 option. Logical Block Addressing (LBA) allows Linux to see beyond the 1024th cylinder on your hard drive. It's active by default on most computers and is detected automatically by GRUB. Even if it isn't active, it doesn't matter as long as the partition with your* `/boot` *directory is located below that cylinder limit.*

FIGURE 4.33
Selecting a bootloader

FIGURE 4.34
Here's your chance to add special kernel parameters.

25. If you're using GRUB, you can set a password. This prevents users with physical access to your computer from booting it in single-user mode to change your root password. This is an excellent idea. Activate the Use A GRUB Password option, and enter the password of your choice, as shown in Figure 4.35. Then press F12 to continue.

NOTE *If your passwords don't match, you'll see a warning to that effect. After pressing Enter, you're then taken to the original screen where you can try to set your password again.*

FIGURE 4.35 Creating a GRUB password

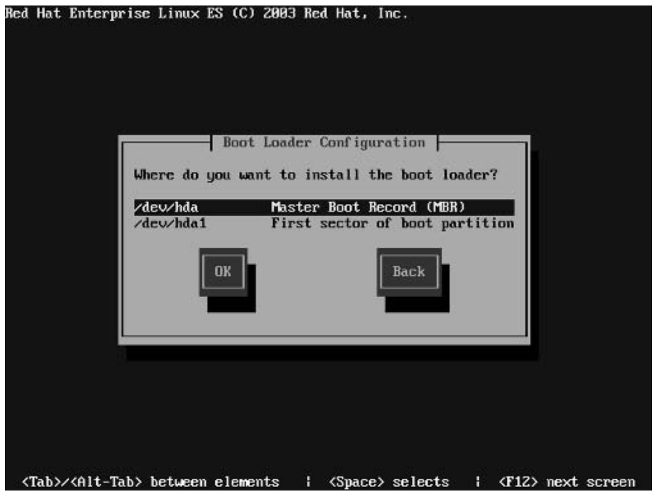
26. Next, you can select the default operating system for your computer. If you're using only Red Hat Enterprise Linux on your computer, this doesn't matter; you get only one choice. However, if Anaconda detects more than one operating system on your computer, you get to select a default in the screen shown in Figure 4.36. But given the nature of Red Hat Enterprise Linux, I'm assuming you'll use it as the only operating system on your computer.

FIGURE 4.36 Selecting a default operating system

For more information on how this works in GRUB, see Chapter 11. If you have more than one operating system on this computer, select the default operating system of your choice, use the Tab key to select OK, and then press Enter to continue.

27. Now you can set the location of your bootloader. Typically, it should be placed on the Master Boot Record (MBR) of your hard drive. However, if you already have a different bootloader on your computer, you may want to choose First Sector Of Boot Partition, which corresponds to the partition with the `/boot` directory. Figure 4.37 shows typical choices. Make your selection, and press F12 to continue.

FIGURE 4.37 Setting a location for your bootloader



NOTE *If you have more than two hard drives, be careful. The BIOS on a typical PC can find the* `/boot` *directory only if it's on one of the first two IDE (*`hda` *or* `hdb`*) or SCSI (*`sda` *or* `sdb`*) hard drives. The SCSI drive must have an ID Number of 0 or 1. If you have both IDE and SCSI hard drives, it must be on the first one of these drives; the SCSI drive must have an ID number of 0.*

28. Now you can complete your network configuration. Since you're installing from a network server, you've already entered basic IP address information for this computer. If you're satisfied with the settings shown in Figure 4.38, press F12 to continue. (Note that the configuration in this case is for the network card labeled `eth0`.) Otherwise, deselect Use BOOTP/DHCP, which allows you to change the settings shown in Table 4.7.

NOTE *If you have more than one network card, you get to repeat step 28 with the next card.*

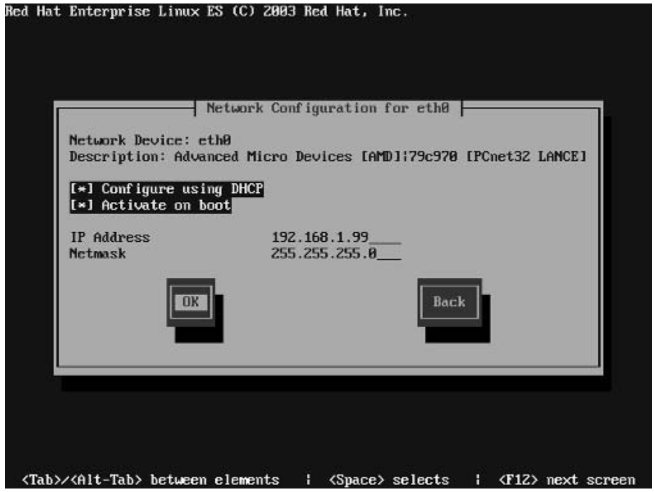


FIGURE 4.38 Configuring network settings

29. If you set a static IP address, you'll now name the IP address of your gateway and DNS servers. You'll then see a Hostname Configuration window, where you can assign a hostname for your computer. Do so, and press F12 to continue.

30. Next, you can configure a firewall for your computer. It's generally not necessary for computers inside a LAN that's already protected by a firewall. However, if you're installing Red Hat Enterprise Linux on a computer that's also connected to another network such as the Internet, a firewall is critical. As shown in Figure 4.39, you can either activate or deactivate a firewall. You can also customize any firewall that you activate. For the purpose of this installation, we'll configure a firewall.

FIGURE 4.39 Configuring the firewall

Table 4.7: Network Configuration Settings (some are shown only if you deselect the Configure Using DHCP option)

Setting	Description
Use BOOTP/DHCP	Makes the computer look for a DHCP server on a local or remote network; the BOOTP protocol makes it possible to get IP address information from a DHCP server on a remote network.
Activate On Boot	Sets the computer to activate this network configuration when you start Linux.
IP Address	Configures the IP address associated with this network card; for more information on IP addressing, see Chapter 15.
Netmask	Short for network mask or subnet mask; for more information, see Chapter 15.
Default Gateway (IP)	Notes the IP address of the computer or router that's also connected to an external network such as the Internet.
Primary Nameserver	Configures a DNS server for this network; the IP address can be outside your LAN.
Secondary Nameserver	Configures a DNS server for this network; the IP address can be outside your LAN.
Tertiary Nameserver	Configures a DNS server for this network; the IP address can be outside your LAN.

Note *Even the standard Red Hat high-security firewall allows the computer to get information from a DNS server, which is essential for browsing the Internet. This holds true as long as you've listed the IP address for at least one* `nameserver` *earlier in the installation process.*

31. Now we'll customize the firewall. Alternatively, you can customize the firewall after installation using the techniques described in Chapter 17. After selecting enable, use the Tab key to highlight Customize, and then press Enter. That opens the Firewall Configuration - Customize screen, shown in Figure 4.35. The options shown in Figure 4.40 are described in Table 4.8. Make the desired changes, and press F12 to return to the basic Firewall Configuration screen in Figure 4.39. Press F12 again to continue.

Table 4.8: Firewall Configuration Customization Options

Option	Description
Trusted Devices	Lets you activate a network card as a trusted device; this is important if you have more than one network card. Any firewall you create won't stop any traffic that passes through that trusted device. It's common to activate this option for a network card connected to an internal network.
SSH	Permits Secure Shell access; you're allowing encrypted remote connections using this service, as described in Chapter 18.
Telnet	Permits Secure Shell access; you're allowing clear-text remote connections using this service, as described in Chapter 18.

Table 4.8: Firewall Configuration Customization Options *(continued)*

Option	Description
WWW (HTTP)	Allows incoming requests to a web server on your network.
Mail (SMTP)	Allows incoming requests to an outgoing e-mail server on your network.
FTP	Permits incoming requests to an FTP server on your network.
Other Ports	You can allow data in through one of the other TCP/IP ports described in `/etc/services`; the format is service:protocol, such as `time:tcp,time:udp`.

Figure 4.40
Customizing a firewall

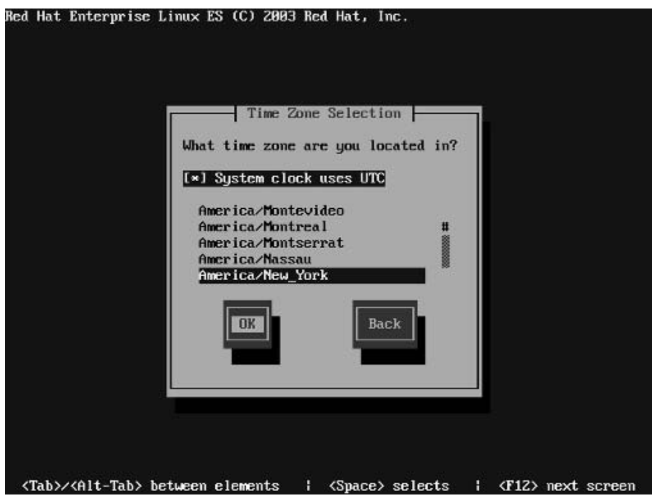
32. In this step, you can customize the language packages installed with Red Hat Enterprise Linux. As you can see in Figure 4.41, English (USA) is installed by default. Support for more than 120 different languages and or dialects is available. If you choose Select All, support for all languages will be installed. If you choose Reset, support for only English (USA) will be installed. Make your choices, and press F12 to continue.

Note *If you select more than one language, you'll see a Default Language screen, where you select the default language for your system.*

33. Next, you can select the hardware clock settings and time zone for your computer. The options are shown in Figure 4.42. If you select System Clock Uses UTC, you should set the clock in your PC's BIOS to Greenwich Mean Time. In addition, you need to select the time zone most closely associated with your location.

FIGURE 4.41
Selecting supported languages

FIGURE 4.42
Selecting a time zone



NOTE *The advantage of activating the System Clock Uses UTC option is that Linux can then adjust your clock for daylight saving time. However, if you have another operating system on your computer, such as Microsoft Windows, this setting would alter that clock. UTC is a French acronym that essentially refers to Greenwich Mean Time.*

34. Now you need to select a root password for your system. This is the password you'll use to log into the root or superuser account. Make your selections, as shown in Figure 4.43, and then press F12 to continue.

FIGURE 4.43
Creating a root password

35. Even if you're installing from one of the Red Hat Enterprise Linux server CDs, you'll see the Workstation Defaults screen shown in Figure 4.44. The defaults, which aren't shown in the figure, include the following package groups: GNOME Desktop Environment, Graphical Internet, Text-Based Internet, Administration Tools, Server Configuration Tools, Web Server, and Windows File Server. You can accept these defaults and change them later using the Red Hat Package Management utility (`redhat-config-packages`) described in Chapter 10.

FIGURE 4.44
Do you accept the default installation environment?

36. It's time to select the package groups that will be installed, as shown in Figure 4.45. You can further customize your installation by selecting or deselecting the package groups of your choice. Chapter 5 describes each package group in more detail.

FIGURE 4.45
Selecting package groups

NOTE *The package groups you install aren't final; you can always install individual packages using the* `rpm` *command described in Chapter 11 and the Red Hat Package Management utility (*`redhat-config-packages`*) described in Chapter 10.*

37. You get one last chance to stop before Anaconda starts installing Red Hat Enterprise Linux on your system, as shown in Figure 4.46. The cited file, `/root/install.log`, will include a complete list of installed packages. If you're ready, highlight OK and press F12 to continue.

Now Anaconda formats your selected partitions. It may take several minutes to start the installation process. Figure 4.47 shows the installation process in action. You can track the current status of the installation.

Once the installation process is complete, Anaconda performs a postinstall configuration automatically. Depending on the speed of your network and the number and size of packages you've installed, the entire installation process could take a few minutes or several hours. Finally, we're 'ready for the next step.

38. Now we're moving into the home stretch—configuring the graphics system. The Video Card Configuration menu, shown in Figure 4.48, illustrates what Red Hat Enterprise Linux was able to detect for your video card. If you don't want to configure your graphics system at this time, you can select Skip X Configuration and press Enter. You'll be taken to step 42.

FIGURE 4.46 You're ready to install Red Hat Enterprise Linux.

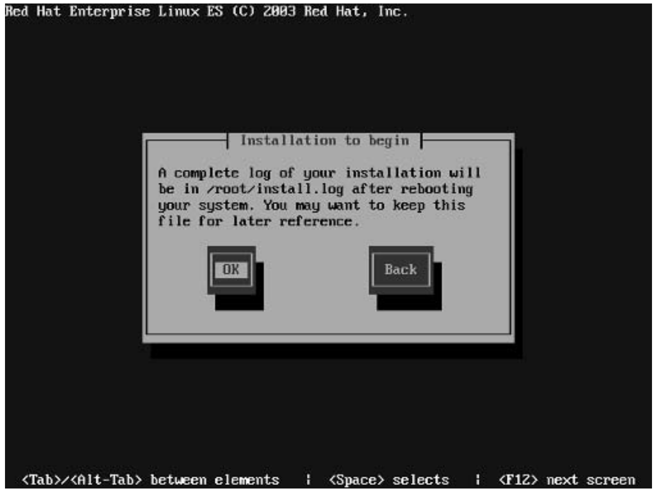
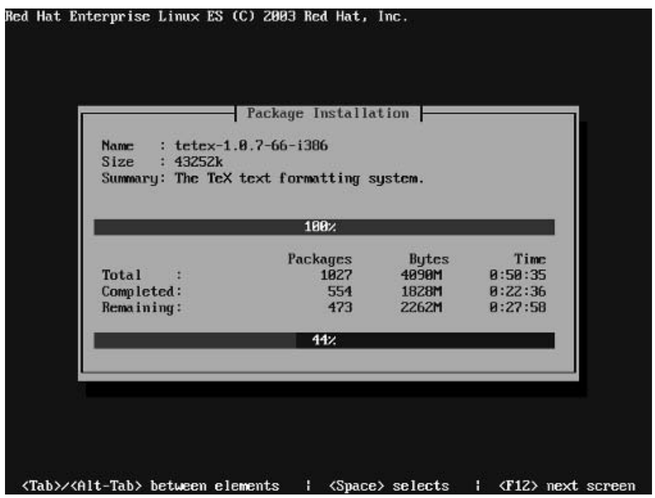


FIGURE 4.47 The installation process at work



39. If you're not satisfied with the current configuration, you can change it. For example, you can highlight the Change option for the video card and press Enter. This opens the Video Card menu, shown in Figure 4.49.

There are a whole series of video cards available; most are proprietary. If you can't find the make and model for your card, you can try a different card from the same manufacturer, or you can select Unsupported VGA Compatible or VESA Driver (Generic). These options correspond to default VGA and SVGA drivers. Select an appropriate driver and press F12 to continue.

FIGURE 4.48
Configuring your video card

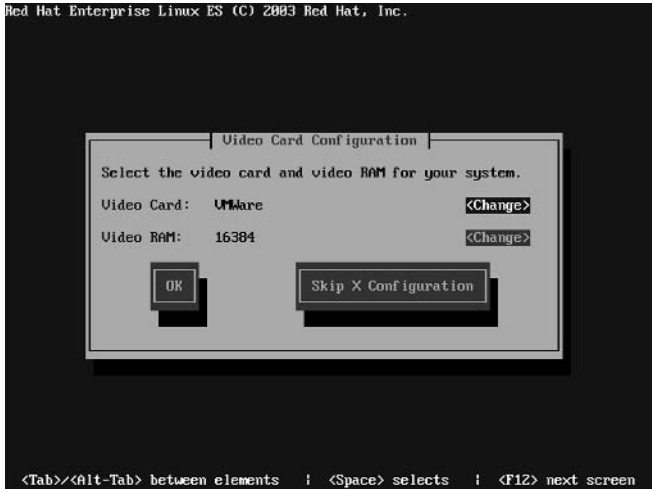
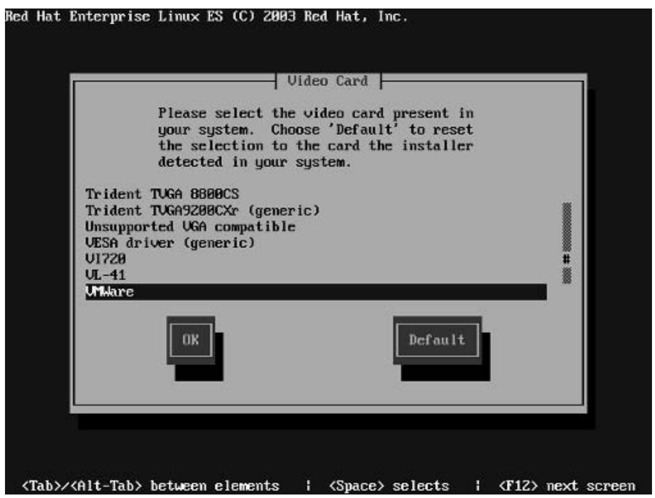


FIGURE 4.49
Selecting a video card



NOTE *VGA is short for Video Graphics Array, the basic color system for most monitors built today. One slightly higher but still common standard is SVGA, short for Super VGA. The Red Hat SVGA driver is VESA, which is short for the Video Electronics Standards Association; VESA also represents the group of standards associated with SVGA.*

40. Red Hat Enterprise Linux may not detect all the RAM associated with your video card. In that case, you can highlight the Change option associated with Video RAM and press Enter. This opens the Video RAM menu, shown in Figure 4.50. Select the amount of Video RAM

associated with your video card, and press F12 to return to the Video Card configuration menu. If you're satisfied with the overall configuration, press F12 again to continue.

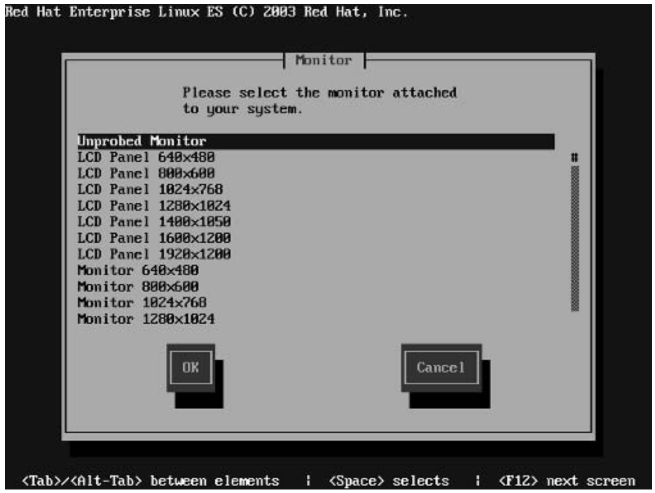
41. When you configure a video card, you also need to configure a monitor. In the Monitor Configuration screen, shown in Figure 4.51, you can specify a make and model for your monitor, as well as the allowable horizontal (HSync Rate) and vertical sync rates (VSync Rate).

FIGURE 4.50 Specifying your - Video RAM

FIGURE 4.51 Configuring your monitor

42. If you're not satisfied with the current configuration, you can change it. It's certainly appropriate if you see that Anaconda labels your system with an "Unprobed Monitor." You can highlight the Change option for the Monitor and press Enter. This opens the Monitor menu, shown in Figure 4.52.

FIGURE 4.52
Specifying a monitor



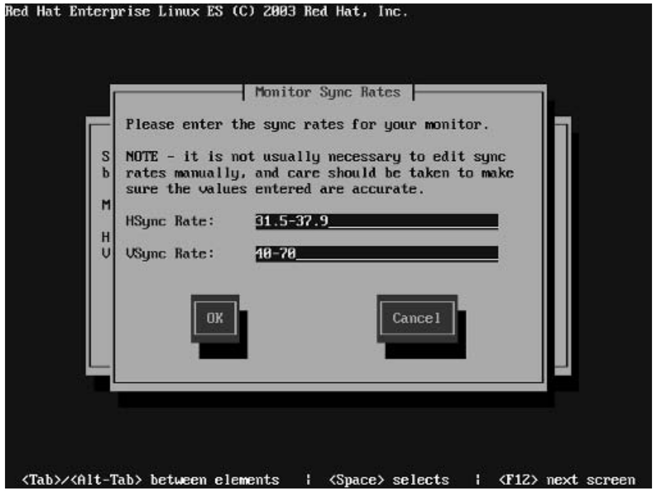
As you can see, there are a whole series of monitors; most are proprietary. If you can't find the make and model for your monitor, you can try a different card from the same manufacturer. In addition, a substantial number of generic monitors are available, including several for laptop computers. Select an appropriate monitor and press F12 to return to the Monitor Configuration window.

43. Next, you'll get to change allowable monitor sync rates, as shown in Figure 4.53. The horizontal sync rate is the amount of time it takes for your system to redraw one horizontal line on your screen; typically it's listed in KHz. The vertical sync rate is the amount of time it takes for your system to redraw the entire screen; typically that is listed in Hz. If you want to make a change, highlight the Change option associated with the HSync Rate and VSync Rate options and press Enter.

WARNING *Be careful! Where possible, check the documentation for your monitor. If you specify horizontal or vertical rates that are too high, signals from the video card could permanently damage your monitor.*

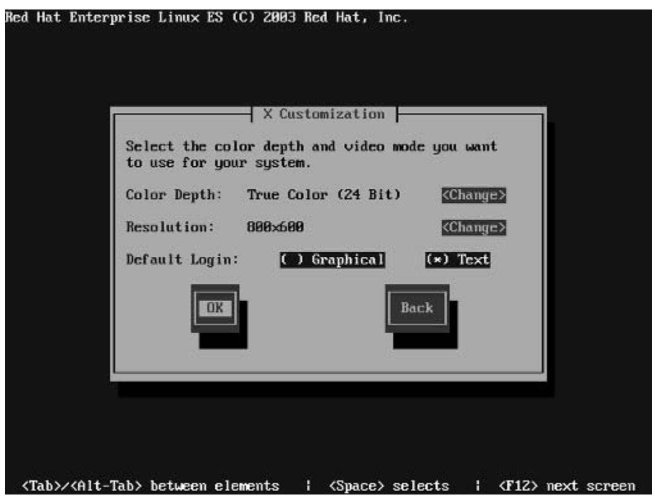
Change the rates as desired, and press F12 to return to the Monitor Configuration menu. If you're satisfied with the overall configuration, press F12 again to continue.

FIGURE 4.53
Setting monitor sync rates



44. Next, you'll get to specify several defaults. As shown in the X Customization menu in Figure 4.54, you can specify a color depth, resolution, and default login mode.

FIGURE 4.54
Selecting graphical startup defaults



45. If you're not satisfied with the current configuration, you can change it. For example, you can highlight the Change option for Color Depth and press Enter. This opens the Color Depth menu, shown in Figure 4.55. Select the color depth of your choice, and press F12 to return to the X Customization menu.

FIGURE 4.55
Selecting a color depth

46. Now you can change the resolution, as shown in Figure 4.56. Highlight the preferred default resolution of your choice and press Enter.

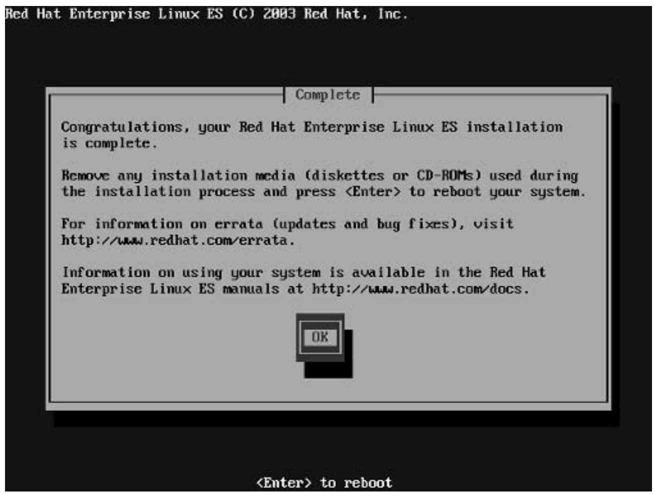
FIGURE 4.56
Specifying a monitor resolution

Available resolutions are based on the memory in your video card, your specified color depth, and the specifications of your monitor. Select the resolution of your choice, and press F12 to return to the X Customization menu.

47. Also in the X Customization menu, you can change the Default Login mode for your Linux system. This changes the default runlevel in your `/etc/inittab` file, described in Chapter 11. You can set up a login at the text or graphical consoles. For a view of available graphical consoles, see Chapter 29. Make any desired changes and press F12 to continue.

48. Finally, installation and preliminary configuration are complete. You should see the screen shown in Figure 4.57. When you press Enter, Anaconda reboots your computer. The next thing you should see after reboot is the bootloader you selected during the process.

FIGURE 4.57 Installation is complete.



NOTE *If you configured a login at the graphical console in step 47, Red Hat Enterprise Linux reboots and starts the* `firstboot` *utility. The steps and views are identical as described in the section on the Red Hat Setup Agent in Chapter 3.*

Text-Mode Upgrades

It's also possible to upgrade a previous version of Red Hat Enterprise Linux. If Anaconda detects the previous version, you'll see the System To Upgrade menu. We looked at it briefly earlier, in step 14 of the full installation process. For your convenience, we'll review the menu here in Figure 4.58. We won't go through the full step-by-step process, since many of the steps are similar to the previously described text-mode installation.

You can also use this technique to install the quarterly update CDs that you get with a paid Red Hat Enterprise Linux subscription. Just start with the `text upgradeany` command at the first installation boot screen.

In this case, Anaconda has found the `/boot` directory associated with Red Hat Enterprise Linux 2.1 installed on the `/dev/hda2` partition. If you simply want to upgrade your packages to Red Hat Enterprise Linux 3, select this option and press F12 to continue.

FIGURE 4.58 The System To Upgrade screen

NOTE *Even though Red Hat supports upgrades from Red Hat Enterprise Linux 2.1, it recommends you install a fresh copy of Red Hat Enterprise Linux 3. While upgrades from Red Hat Linux are possible, they aren't supported. Furthermore, you're not allowed to customize the packages to be upgraded.*

The screen that follows allows you to upgrade or install a new bootloader. As shown in Figure 4.59, you have three choices, which are described in Table 4.9. If you have a special bootloader configuration file, such as one with special kernel entries, you may not want to wipe it out with the Create New Boot Loader Configuration option.

FIGURE 4.59 Upgrading the bootloader

Table 4.9: Choices in Upgrading a Bootloader

Option	Description
Update Boot Loader Configuration	Upgrades the bootloader package without changing the configuration file.
Skip Boot Loader Updating	Doesn't upgrade the bootloader package and doesn't change the configuration file.
Create New Boot Loader Configuration	Installs a new bootloader package; you'll have to change the configuration file.

Select the option of your choice, For the purpose of this chapter, we'll select the first option. and press F12 to continue. Assuming that you're customizing packages to be upgraded, Anaconda proceeds to read the current packages on your system.

NOTE *If you're upgrading with a Red Hat Enterprise Linux update CD, all packages are updated, including the Linux kernel. Unlike the technique we recommend in Chapter 12, this upgrade technique replaces the existing kernel.*

Troubleshooting a Network Installation

If you're unable to install Red Hat Enterprise Linux over a network, there are a number of things that you can check. First, most network problems are physical. Red Hat Enterprise Linux installs a firewall by default, and that firewall may also cause problems. If you don't have correct address settings, your computer won't be able to find the installation server. Finally, there are special issues related to network installations of Red Hat Enterprise Linux on a laptop.

Checking the Messages

When you're having a problem with a network installation of Red Hat Enterprise Linux, the problem may not even be with the network. Several text installation screens can provide valuable messages. You can get to these screens with the Ctrl+Alt+F*n* command, where *n* is a number between 2 and 5. I've described these screens in Table 4.10.

Table 4.10: Red Hat Installation Screens

Screen	Description
Ctrl+Alt+F1	Returns to the main installation screen.
Ctrl+Alt+F2	Opens a bash shell with limited command capabilities; for example, the `df` command can show mounted directories and partitions. Other bash commands are described in Part II of this book.Navigate to the `/mnt/sysimage` directory; you can begin to configure your system as it's being installed.
Ctrl+Alt+F3	Views the installation log, with messages related to hardware detection; if you're having trouble reading CDs or loading drivers, check here.

Continued on next page

TABLE 4.10: RED HAT INSTALLATION SCREENS *(continued)*

SCREEN	DESCRIPTION
Ctrl+Alt+F4	Goes to the system message log, with messages such as formatting and mounting directories on partitions.
Ctrl+Alt+F5	Notes various messages such as filesystem labels, blocks, formats, and journals.

Checking the Network

Again, most network problems are physical. When you have a condition where your network isn't working, that inevitably means you end up checking your cables, connections, and other hardware components.

There are a number of commands that you can use to test physical connections as well. For example, when you use the `ping` command on the IP address of another computer, you're testing the connectivity between your computers. Basic network troubleshooting techniques are described in Chapter 16.

The Firewall on the Server

Red Hat Linux and Red Hat Enterprise Linux both install a firewall by default. However, if you accept the default installation, you won't be able to access installation files from that computer, at least using the network protocols (NFS, HTTP, and FTP) discussed in this chapter. If you're still having network problems, it's worth logging into and checking the server computer with the Red Hat Enterprise Linux installation files.

If that server is also running Red Hat Linux 7.3 and above, there are a couple of simple commands that you can use to check for a firewall:

```
# iptables -L
```

This `iptables` command lists any current rules that apply to that computer. If there are existing rules, you can "flush" them from the current configuration with the following command:

```
# iptables -F
```

When the installation is complete, you can reenable the firewall with the `service iptables restart` command. Firewalls are covered in more detail in Chapter 17.

Address Settings

Users who are less familiar with IP addressing may make mistakes. For example, if you've set a static IP address for your computer, you need to make sure you have several things right:

- The IP address of the computer should be on the same network as your LAN.
- The network mask (or subnet mask) of the computer should match that of every other computer on the LAN.

While it's useful to have the correct default gateway and DNS server IP addresses, those aren't absolutely necessary for a successful network installation of Red Hat Enterprise Linux. For more information on how IP addressing works, see Chapter 15.

Summary

In this chapter, we installed Red Hat Enterprise Linux over a network. You need a second computer to hold the installation files. We looked at the process for configuring three network services: NFS, HTTP, and FTP. The step-by-step process is fairly straightforward. Details of each service are described in later chapters.

We examined the tools available for setting up a PXE boot server, which allows you to set up automated installations of computers with PXE network cards. When you boot these computers, they detect a DHCP server and use the Kickstart file that you configure to install Red Hat Enterprise Linux automatically.

Then we looked at the various options for boot and driver disks. We then looked at the network installation process, in text mode, in a step-by-step fashion. We also took a brief look at the upgrade process.

If you have trouble with a network installation, you may be able to get clues to the problem using the installation message screens. If you have a network problem, most problems are physical; there are also a number of commands that you can use to inspect your network. One common mistake is to leave an active firewall on the installation server, which can block communication. Another common mistake is based on errors in IP address information. Finally, special problems can occur when you install Red Hat Enterprise Linux on a laptop computer.

In the next chapter, we'll look at automating the installation process with Kickstart. Once you've set up a Kickstart file, you'll be able to install Red Hat Enterprise Linux on a several computers simultaneously, using the same network installation server.

Chapter 5

Kickstarting Linux

In previous chapters, you learned to install Linux from local and remote sources. There are many ways to customize Red HatEnterprise Linux; all require extensive user input. Thus, if you're an administrator responsible for installing Red Hat Enterprise Linux on a group of computers, you could spend a lot of time installing and customizing Linux on every last computer. For this reason, Red Hat has developed the Kickstart system to automate the installation process. With it, you can manage the installation of package groups, or even individual RPM packages, on each of your computers.

As you've seen in Chapters 3 and 4, packages are collected together in groups such as GNOME Desktop Environment and Graphics. These package groups are organized in the `comps.xml` file on the first Red Hat installation CD, in the `/RedHat/base` directory. We'll examine this file in detail; you can edit the file to customize how your users install Red Hat Enterprise Linux.

The software in some packages and package groups won't work unless other software is installed. These are known as *dependencies*, which are also documented in `comps.xml`.

When you install Red HatEnterprise Linux 3 on your computer, Anaconda leaves a default Kickstart file, `anaconda-ks.cfg`, in your `/root` directory. You can use this file to create a standard Kickstart file for your other computers. In addition, Red Hat includes the detailed GUI Kickstart Configurator, which can help you customize the Kickstart file you need.

Once you've created a Kickstart file, you can set it up on a boot disk or network source. All you need to do is reboot your computer with the boot disk. Once the basic kernel is loaded, it can get Red Hat Enterprise Linux installation files locally or through your network. This chapter covers the following topics:

- Grouping packages: comps.xml
- Analyzing the default Kickstart configuration
- Working with the GUI Kickstart Configurator
- Kickstarting from a boot disk

Grouping Packages: *comps.xml*

Anaconda, the Red Hat Enterprise Linux installation program, uses the `comps.xml` file to set up your installation. This file is located on the first installation CD, or the network source, in the `/RedHat/base` directory. It's written in XML, which is primarily used for web pages. It includes tags that are functionally similar to the standard language of web pages, HTML.

NOTE *Once Red Hat Enterprise Linux is installed (on an x86 PC), you can also find* `comps.xml` *in the* `/usr/share/comps/i386` *directory. If possible, open this file in a text editor to follow the descriptions in the first part of this chapter.*

The `comps.xml` file includes four basic sections. First are mandatory package groups that are normally installed with every Red Hat Enterprise Linux installation. Then you have individual package groups, which you can select during the installation process. Third, these groups are organized in categories, which you can see during the graphical installation process or through the `redhat-config-packages` utility described in Chapter 10. Finally, there is a list of dependencies, which are packages required by others.

Once you understand `comps.xml`, you can edit this file. For example, you can add stanzas with your own special package groups. You can also delete or hide stanzas that you don't want users to install on their computers, such as Graphics or Games.

NOTE *Red Hat Enterprise Linux is organized through the Red Hat Package Manager (RPM). Red Hat software is collected together in RPM files, which end with* `.rpm`*. When RPM packages are collected together in* `comps.xml`*, they are organized in package groups.*

Basic *comps.xml* Stanzas

There's a standard organization to each stanza in the `comps.xml` file. Like HTML, each stanza is enclosed with a starting tag such as `<group>` and an ending tag such as `</group>`. Each group has an identifier, as in the following:

```
<id>dialup</id>
```

Next, these lines determine whether a user is allowed to select the group, and whether it's installed by default:

```
<uservisible>false</uservisible>
<default>true</default>
```

This combination means that this particular group is installed by default. Since this package group isn't visible to the user during the installation process, Anaconda will automatically install it.

If you don't see either tag, `<uservisible>` is true and `<default>` is false. In other words, by default, package groups are visible but not selected during the installation process.

Each group includes a name and a description; the `comps.xml` file includes versions of the following lines in different languages. These commands list the name of the KDE package group, the name in German, and an abbreviated description in the same language:

```
<name>KDE Desktop Environment</name>
<name xml:lang="de">KDE Desktopumgebung</name>
<description>KDE ist eine leistungsstarkes</description>
```

Some groups depend on others. For example, the Graphics group depends on the `base` and `base-x` groups, as documented by the following commands (we omitted most of the commands in this stanza for clarity):

```
<group>
   <id>graphics</id>
   <name>Graphics</name>
   <grouplist>
      <groupreq>base</groupreq>
      <groupreq>base-x</groupreq>
   </grouplist>
</group>
```

Finally, certain packages are associated with each group, as delineated by the `<packagelist>` tag. Some packages are mandatory, meaning that the package can't function without them. For example, the Windows File Server package group can't function without the `samba-client` and `samba` RPM packages.

```
<packagelist>
     <packagereq type="mandatory">samba-client</packagereq>
     <packagereq type="mandatory">samba</packagereq>
     <packagereq type="default">redhat-config-samba</packagereq>
</packagelist>
```

Other packages are classified as default or optional. Users who select individual packages during the installation process can select or deselect these packages.

Some packages are listed as part of multiple package groups. For example, `redhat-config-samba` is part of the Windows File Server and Server Configuration Tools package groups.

NOTE *Red Hat does not include lists of dependencies in the Enterprise version of the* `comps.xml` *file. If you're interested, this information is available in the source code for individual packages.*

Mandatory Groups

There are three mandatory package groups in comps.xml: Core, Base and, Dialup Networking Support. The Core group includes RPM packages that Linux can't live without; some of these packages are listed in Table 5.1. You can review the full list for yourself at the top of the `comps.xml` file.

TABLE 5.1: SOME CORE LINUX PACKAGES

PACKAGE	DESCRIPTION
`basesystem`	The first package installed in Red Hat Enterprise Linux; it should never be deleted.
`bash`	The Bourne Again Shell; it's the default Red Hat Enterprise Linux command interpreter.
`cpio`	An archiving utility; see Chapter 14.
`e2fsprogs`	The basic Linux filesystem management commands.
`filesystem`	The standard directory layout.

Continued on next page

Table 5.1: Some Core Linux Packages *(continued)*

Package	Description
`glibc`	Standard C language libraries.
`grub`	The default Linux bootloader; see Chapter 11.
`hotplug`	For USB and IEEE 1394 devices.
`iputils`	A package that includes basic networking commands such as `ping`.
`kbd`	For managing a console, fonts, and the keyboard.
`kernel`	The Linux kernel.
`libgcc`	A package that supports the GNU C language compiler.
`passwd`	A package that includes the `passwd` command.
`procps`	System Information utilities, such as `ps`.
`raidtools`	For configuring a software RAID device.
`rpm`	A package that includes the Red Hat Package Manager; see Chapter 10.
`setup`	Some basic `/etc` configuration files, such as `passwd`, `group`, and `profile`.
`vim-minimal`	The `vi` editor.

The Base group includes RPM packages that make Red Hat Enterprise Linux useful to administrators. A very few of these packages are listed in Table 5.2.

Table 5.2: Some Base Linux Packages

Package	Description
`at`	Supports the `at` and `batch` commands described in Chapter 13.
`bind-utils`	Contains commands for checking DNS (Domain Name Service) servers; see Chapter 19.
`crontabs`	For regularly scheduled jobs; see Chapter 13.
`dhclient`	Contains the DHCP (Dynamic Host Configuration Protocol) client.
`ftp`	Contains the FTP (File Transfer Protocol) command-line client.
`kudzu`	Contains the Red Hat hardware- probing tool.
`nfs-utils`	Contains Network File System (NFS) commands; see Chapter 22.
`openssh-clients`	For SSH (Secure Shell) client connections.
`quota`	Allows you to set quotas; see Chapter 9.

Continued on next page

Table 5.2: Some Base Linux Packages *(continued)*

Package	Description
`sudo`	Lets you configure certain users with root privileges.
`telnet`	Contains the Telnet command-line client.
`up2date`	Contains the Red Hat Update Agent; see Chapter 10.
`ypbind`	Contains the NIS (Network Information Service) client; see Chapter 23.

These package groups together include nearly 600MB of files. The Dialup Networking Support package includes the basic packages associated with telephone modem and ISDN adapter connections. Now let's take a look at the other package groups that we see during the Red Hat Enterprise Linux installation process.

Package Groups

In this section, we'll look at each package group in some detail, based on the `comps.xml` file. It should help you decide on a standard set of software packages to install on your computers.

You may not see all of these groups during the Red Hat Enterprise Linux installation process; as noted earlier, you can configure `comps.xml` to leave out one or more groups from the display.

The order of packages in this section corresponds to the `comps.xml` file available as of this writing. And the order is different from what you see during the Red Hat Enterprise Linux installation process. Some of the package groups you see in `comps.xml` won't even show up during the visible installation process.

Some package groups depend on others. For example, the Office/Productivity package group won't work unless the X Window System package group is also installed. One component of this group, OpenOffice.org, requires installation of a number of other packages.

Core

The Core package group is installed by default. You won't see this package group during the standard installation process. It is "mandatory" and includes packages that are fundamental to the operation of the Linux operating system, as we've described earlier in this chapter.

Base

The Base package group is installed by default. You won't see this package group during the standard installation process. It is "mandatory" and includes packages associated with basic operation of the Linux operating system, as we've described earlier in this chapter.

Printing Support

It may seem strange to have the Printing Support package group this early in the `comps.xml` file. The fonts associated with this package are required for the GUI. Naturally, Printing Support also includes basic drivers and utilities associated with the CUPS service described in Chapter 20. This group is installed by default.

X Window System

The X Window System package group includes the XFree86 server and associated packages required to configure a basic GUI on your Linux computer. It includes some basic `redhat-config-*` utilities for managing the date, the network, the service runlevel configuration, sound hardware, users, printers, and, of course, the X Window.

You need this group if you want to install a desktop such as GNOME or KDE. It's installed by default, and requires the Printing Support package group, primarily for its fonts.

Other commands in `comps.xml` may refer to this group by its ID; for example, the following command refers to the `<id>` of the X Window System:

```
<id>base-x</id>
```

Dialup Networking Support

The Dialup Networking Support package group includes the basic utilities required to make a connection via telephone modem or ISDN adapter. You won't see this package group during the standard installation process. Other GUI Internet connection utilities depend on this package (see Chapters 29 and 30). This package group is always installed as a part of Red Hat Enterprise Linux 3.

GNOME Desktop Environment

The GNOME Desktop Environment package group contains the software you need to run the GNOME desktop. It includes basic applications such as text editors, graphical utilities, and more.

You should install GNOME or KDE for users who want a GUI desktop environment. It won't work unless you install the X Window System package group. GNOME is the default Red Hat Enterprise Linux desktop, and we cover it in Chapter 30.

KDE Desktop Environment

The KDE Desktop Environment package group contains the software you need to run the KDE desktop. It also includes basic applications such as text editors and administrative utilities. And like the GNOME Desktop Environment package group, it won't work unless you install the X Window System package group. KDE, which is the most popular desktop for other Linux distributions, is covered in Chapter 30.

Graphical Internet

The Graphical Internet package group includes basic GUI utilities associated with Internet connections. These include the Mozilla web browser and the Evolution mail manager, as well as several chat and related utilities. While this is a book focused more on servers, some of these utilities are described in Chapter 30.

Text-Based Internet

There are a number of handy utilities that you can use to connect to the Internet from a text console. For example, `elinks` is a web browser with a surprising array of features, and `mutt` is a competent e-mail client.

Sound and Video

This is an all-in-one package group for controlling, configuring, and commanding a sound card. It includes several tools for recording multimedia or data on CDs and DVDs.

Graphics

The Graphics package group includes several utilities for managing pictures, screenshots, and other graphics. This includes The GIMP and associated data, which is briefly covered in Chapter 30. I've used The GIMP extensively to create screenshots for this book.

If you want graphics, naturally you'll need the X Window System package group.

Office/Productivity

This package group includes the fully featured OpenOffice.org office suite. It also includes office-style applications associated with GNOME Office and KOffice, plus a couple of other applications, such as a project manager, in the same category. The OpenOffice.org suite is briefly described in Chapter 30.

Mail Server

This package group includes the sendmail and Postfix mail servers. Optional packages can help you set up Web-based email, filter unwanted e-mail, and more. For more information on the sendmail and Postfix mail servers, see Chapter 21.

Network Server

The Network Server package group includes a variety of servers that can be useful for managing a LAN. Available servers range from DHCP (for managing IP address information) to `krb5-workstation` (which includes a Kerberos capable Telnet server). More information on these servers can be found in Chapters 18, 19, and 22.

Legacy Network Server

The Legacy Network Server package group includes several older servers. Red Hat discourages the use of the RSH and regular Telnet servers for security reasons. However, they are still in common use, and can be relatively safe within an internal private network.

News Server

The News Server package group consists of only one package, InterNetNews (`inn`). This server allows you to set up a news server similar to Usenet discussion list servers that you can access through some mail managers.

Windows File Server

The Windows File Server package group is also fairly simple; all you need is the `samba` and `sambaclient` packages to connect to and share with other computers on a Microsoft Windows–based network. The `redhat-config-samba` tool is useful for configuring basic shares from the GUI. Samba is covered in Chapter 24.

Server Configuration Tools

Red Hat has recently created several configuration tools, starting with `redhat-config-*`, where `*` represents the function. This package group allows you to use these tools to configure a number of servers. Although it isn't specified in `comps.xml`, most of these tools won't work unless you're running an X Window interface.

Note *One tool that does work without X Window is* `redhat-config-xfree86`*, which creates its own GUI even from the regular command-line interface.*

FTP Server

The FTP Server package group is straightforward. It includes one package, the Very Secure FTP Daemon. It allows you to set up an FTP server with a decent level of security. We cover FTP configuration in Chapter 22.

SQL Database

The SQL Database package group allows you to run a PostgreSQL database server, which uses the Structured Query Language (SQL) and is a relational database server. While we don't cover PostgreSQL, we cover MySQL in Chapter 26.

MySQL Database

The MySQL Database package group allows you to run a MySQL relational database server, which we cover in Chapter 26.

Web Server

The Web Server package group includes two different web servers, Apache (`httpd`) and Tux, which are discussed in Chapter 25. This package group also includes a number of Apache modules.

DNS Name Server

The DNS Name Server package group includes two packages: `bind` is the standard DNS server on Linux, and the `caching-nameserver` package supports a DNS server cache on a computer. DNS is covered in more detail in Chapter 19.

Authoring and Publishing

The Authoring and Publishing package group covers Linux's native publishing format, DocBook. It's a format for marking up text files that allows you to transform your document into one of several formats, including HTML, RTF, and TeX. The DocBook system is not covered in this book. For further reading, try *DocBook: The Definitive Guide* by Norman Walsh and Leonard Muellner (O'Reilly, 1999).

Engineering and Scientific

The Engineering and Scientific package group includes a series of packages for calculations. Some relate to linear algebra, to help you solve complex equations. Since this is not an engineering book, we won't cover these packages.

Editors

Two of the most popular Linux text editors are part of the Editors package group: `vi` and Emacs. The `vi` editor is actually installed by default; this package group includes the enhanced version of this editor, `vim`. If you install `emacs` or its GUI cousin, `xemacs`, Red Hat Enterprise Linux automatically installs the associated package groups that follow. Entire books have been written about both of these editors; I cover `vi` briefly in Chapter 6.

For more information on Emacs, see the *GNU Emacs Manual* by Richard M. Stallman (GNU Press, 2002).

Emacs

The Emacs group includes the Emacs text editor and a couple of packages for editing the LISP and SGML computer languages. You won't see this package group during the standard installation process.

Xemacs

The Xemacs group includes three packages for making Emacs work within a GUI. You won't see this package group during the standard installation process.

Ruby

While the Ruby package group isn't shown during the standard installation process, it does include the packages associated with the Ruby programming language. It's a scripting language functionally similar to Perl. We do not cover programming languages in this book.

System Tools

The System Tools package group includes a wide variety of client and diagnostic software. For example, as shown in Chapter 17, Ethereal allows you to read clear-text messages on your network. As we explain in Chapter 24, you can use a number of tools associated with `samba-client` to read shared directories on a Microsoft Windows–based network.

Administration Tools

The Administration Tools package group includes those `redhat-config-*` utilities that don't fit into other groups. Naturally, this includes a broad range of tools, from keyboard configuration to user management. We describe these tools in as front ends to text-based tools in various chapters in this book.

Games

Linux has games you can install as part of the GUI. Personally, I don't install them, since I don't want to learn how to play another version of Freecell. However, some administrators think games can help the novice user become more comfortable with Linux. This package group includes games associated with the GNOME desktop.

ISO8859 Support

There are four different ISO8859 font sets that you can install. You won't see any of these package groups during the standard installation process. ISO8859-2 is associated with Eastern European languages. ISO8859-9 is associated with the Turkish languages. ISO8859-14 is associated with the Welsh language. Finally, ISO8859-15 provides Euro support. These packages include fonts at 75 and 100 dpi (dots per inch).

NOTE *ISO is the International Organization for Standardization (`www.iso.ch`). As strange as it sounds, the acronym does not match the official title (nor does it match the French translation of the title).*

Individual Language Support

There are a number of other package groups may allow you to use Linux in your native tongue. You can select the language(s) of your choice before the main screen with package groups. Each individual language group includes fonts; many include spell checkers and translated man pages. As of this writing, support is available for the Cyrillic alphabet, as well as Afrikaans, Brazilian Portuguese, British English, Canadian English, Catalan, Chinese, Czech, Danish, Dutch, Estonian, Finnish, French, German, Greek, Hebrew, Hungarian, Icelandic, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swedish, Syriac (Aramaic), Turkish, and Ukrainian.

NOTE *Some of these languages require different font sets; for example, Ukrainian requires the Cyrillic alphabet package, and Turkish requires the ISO8859-9 package.*

Development Tools

If you do any sort of software development work, you'll need at least some of the packages from the Development Tools package group. While this is not a programming book, you'll need some of these packages to recompile the Linux kernel in Chapter 12.

Prominent packages include `automake`, which allows you to create `Makefile`-style configuration scripts; `binutils`, which includes binary management utilities; and `gcc`, the GNU C language compiler. This package group also depends on the installation of the Development Libraries package group.

Development Libraries

The Development Libraries package group includes many development programs for a wide variety of applications. You won't see this package group during the standard installation process. These libraries range from `kudzu-devel`, which supports the Red Hat hardware management utility, to `openssl-devel`, which lets you configure the SSH server described in Chapter 18. If you're working on improvements to any of these applications, you may need to install this package group.

Kernel Development

If you're planning to modify or reconfigure the Linux kernel, you'll need to install the Kernel Development package group. This group includes the `kernel-source` package; it also depends on the installation of the Development Tools package group. For more information on these packages and managing the kernel, see Chapter 12.

Legacy Software Development

Red Hat has relatively recently upgraded the GNU C language compiler packages. You may still be using software that requires older versions of this package. These legacy packages are organized in the Legacy Software Development package group.

X Software Development

If you're working on the XFree86 software, you may need to install the X Software Development package group. This group includes the packages you need to develop applications for the X Window system. Since there are other desktops, this group doesn't require the software associated with the GNOME or KDE Software Development packages.

GNOME Software Development

If you're developing applications for the GNOME desktop, you'll need to install the GNOME Software Development package group. A couple of key packages include `gtk+-devel`, The GIMP toolkit (GTK+), and `fontconfig-devel`, for managing fonts on your desktop. While GTK+ was created for The GIMP, it's also used to help develop GNOME applications.

KDE Software Development

If you're developing applications for the KDE desktop, you'll need to install the KDE Software Development package group. A couple of key packages include `cups-devel`, for the CUPS print server, and `qt-devel`, for the Qt language toolkit. Qt is the KDE version of the GTK+ toolkit, used to develop KDE applications.

Note *Qt is a C++ language toolkit for creating GUI applications. Developed by Trolltech (*`www.trolltech.com`*), it is not related to QuickTime from Apple. In this case, Qt is not an acronym.*

Package Group Categories

The `comps.xml` file organizes each package group into one of several categories. You've seen how it works in Chapter 3. The standard categories are described in Table 5.3.

Table 5.3: Package Group Categories

Category	Description
Applications	Allows the installation of a variety of package groups, including Graphical Internet, Editors, and Office/Productivity
Desktops	Configures the installation of the X Window and GNOME and/or KDE Desktop environments
Development	Permits you to add various development tool package groups
Servers	Lets you select from several different server package groups, including web, mail, and FTP services
System	Allows you to set up administrative or system tools and/or printing support

For example, the Desktops category, as follows, includes the package groups that you may want or need to install the GUI on your computer:

```
<category>
      <name>Desktops</name>
      <subcategories>
        <subcategory>base-x</subcategory>
        <subcategory>gnome-desktop</subcategory>
        <subcategory>kde-desktop</subcategory>
      </subcategories>
</category>
```

The Desktops category includes `base-x`, `gnome-desktop`, and `kde-desktop`. Based on their `<id>` variables (near the top of the `comps.xml` file), these correspond to the following package groups: X Window System, GNOME Desktop Environment, and KDE Desktop Environment.

Editing Examples

You can help your users customize more during the installation process. For example, you can make package groups such as Core and Dialup Networking visible during the installation process. All you need to do is change the `<uservisible>` variable associated with the specified package group. For example, if you wanted to make the aforementioned packages visible, you'd first change the `<uservisible>` line so it reads as follows:

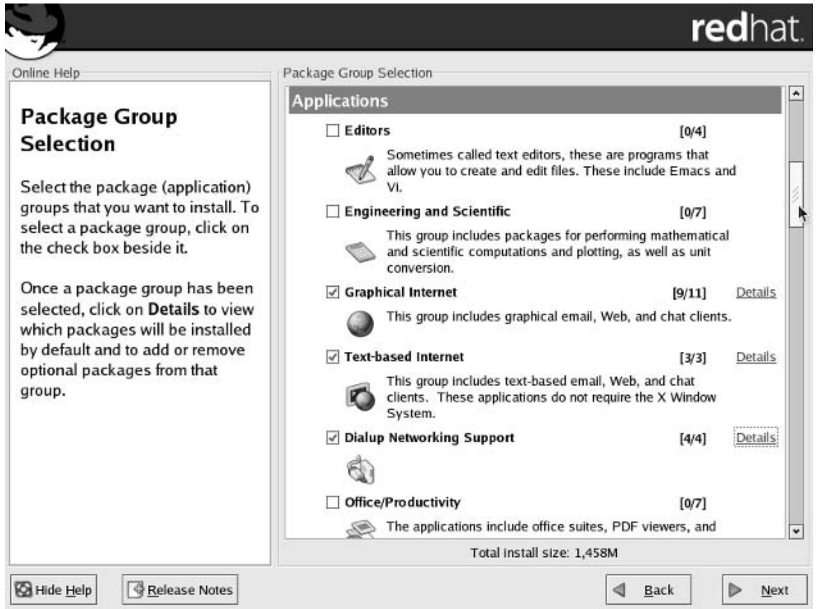
```
<uservisible>true</uservisible>
```

You could then add the package groups of your choice to the appropriate menus, later in the `comps.xml` file. The key is the `<id>` variable. For example, the `<id>` variable for the Dialup Networking Support package group is `dialup`. We've added this variable in bold as a `<subcategory>` to the Applications group listing (I've omitted XML commands specifying other languages):

```
<category>
    <name>Applications</name>
    <subcategories>
        <subcategory>editors</subcategory>
        <subcategory>engineering-and-scientific</subcategory>
        <subcategory>graphical-internet</subcategory>
        <subcategory>text-internet</subcategory>
        <subcategory>dialup</subcategory>
        <subcategory>office</subcategory>
        <subcategory>sound-and-video</subcategory>
        <subcategory>authoring-and-publishing</subcategory>
        <subcategory>graphics</subcategory>
        <subcategory>games</subcategory>
    </subcategories>
</category>
```

The result during the installation process is shown in Figure 5.1.

FIGURE 5.1
Anaconda as modified



Analyzing Your Default Kickstart Configuration

When you install Red Hat Enterprise Linux , the configuration you selected is saved in `anaconda-ks.cfg`, in the `/root` directory. In this section, we'll break down an example of this file from my desktop computer. Figure 5.2 shows the start of this file.

FIGURE 5.2
A typical anaconda-ks.cfg file

```
# Kickstart file automatically generated by anaconda.

install
lang en_US.UTF-8
langsupport --default en_US.UTF-8 en_US.UTF-8
keyboard us
mouse genericwheelps/2 --device psaux --emulthree
xconfig --card "VESA driver (generic)" --videoram 65536 --hsync 31.5-48.5 --vsyn
c 40-70 --resolution 800x600 --depth 24 --defaultdesktop gnome
network --device eth0 --bootproto static --ip 192.168.1.122 --netmask 255.255.25
5.0 --gateway 192.168.1.113 --nameserver 207.217.120.83,207.217.126.81 --hostnam
e RHEnterprise3
rootpw --iscrypted $1$6m2fvmth$sOY4g7ibaSR6uOl4lzUecO
firewall --disabled
authconfig --enableshadow --enablemd5
timezone Etc/GMT-14
bootloader --location=mbr --append hdc=ide-scsi
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
#clearpart --linux --drives=hda
#part /boot --fstype ext3 --size=100 --ondisk=hda
"anaconda-ks.cfg" 52L, 1434C
```

Each Kickstart file can be divided into several categories of commands. We'll look at my `anaconda-ks.cfg` file in the following sections. The order of commands in your Kickstart file may not match what you see here.

Once you've finished editing this file, save it as `ks.cfg`. You'll learn how to set it up on a boot disk toward the end of this chapter.

Preinstallation Commands

You can set up parameters for your installation. For example, you may note the date and time the installation started. The `/etc/motd` file displays each time you log into Linux:

```
%pre
echo "My Kickstart Installation started on `/bin/date`"  >/etc/motd
```

Preinstallation commands should be placed near the end of your Kickstart file, just before any `%post` installation commands you may have.

More extensive scripts are of course possible, but they're limited by the commands available through the disk with the Kickstart file. As you'll see toward the end of this chapter, the Kickstart file is normally copied to the Red Hat installation boot disk, which includes a basic kernel with a limited number of bash shell commands.

Basic Configuration

Only a few basic commands are required to start the Red Hat Enterprise Linux installation process. The following commands are taken from my `anaconda-ks.cfg` file:

```
install
lang en_US.UTF-8
langsupport --default en_US.UTF-8 en_US.UTF-8
keyboard us
mouse genericwheelps/2 --device psaux
timezone America/New_York
bootloader --location=mbr --append hdc=ide-scsi
```

If you're planning to install Linux on a series of other computers, it's best if you're using the same language, keyboard type, and mouse. If that's your situation, you probably won't make any changes. But just in case, let's examine these commands, one at a time.

INSTALL

The first command looks simple; in fact, it's too simple to support an automated installation. In other words, this command doesn't specify the source of the Red Hat installation files:

```
install
```

You could set up Kickstart to look for installation files on your CD or from a hard drive with one of the following options:

```
cdrom
harddrive --partition=hdb1 --dir=/mnt/inst
```

For the purpose of this section, let's assume that the `/RedHat` installation directory is part of the `/mnt/inst` directory and that the server has an IP address of 192.168.0.1.

The `harddrive` command looks for the `/RedHat` directory on the second IDE hard disk on the local computer, on the first primary partition (`hdb1`), in the `/mnt/inst` directory. Or you could install from an NFS shared directory from the remote computer with the following command:

```
nfs --server=192.168.0.1 --dir=/mnt/inst
```

Alternatively, you can install from Red Hat installation files on a remote FTP or web server, using one of the following commands:

```
url --url ftp://username:password@192.168.0.1/mnt/inst
url --url http://192.168.0.1/mnt/inst
```

If you're installing from an anonymous FTP server, the username and password are not required.

LANG AND *LANGSUPPORT*

The next commands specify the language to use during the installation process, as well as the language files to install with Red Hat Enterprise Linux. For example, the following command installs Red Hat Enterprise Linux using standard U.S. English:

```
lang en_US.UTF-8
```

A number of other language codes are available; you can find a list in the `locale.alias` file in the `/usr/X11R6/lib/X11/locale` directory. If you're running an automated installation, you probably won't see any of the installation screens, anyway. However, to install U.S. English as the language you see when you start Red Hat Enterprise Linux, use the following command:

```
langsupport --default en_US.UTF-8 en_US.UTF-8
```

Other available languages include French (`fr_FR`), German (`de_DE`), and Korean (`ko_KR.eucKR`). The language you installed on your computer should be shown in your `anaconda-ks.cfg` file. You can choose from several other languages; check the Red Hat Enterprise Linux System Administration Guide, which is available on the Red Hat documents CD or online at `www.redhat.com`.

NOTE *To get to the online Red Hat Enterprise Linux 3 manuals, navigate to* `www.redhat.com/docs/manuals/enterprise/`.

KEYBOARD

The `keyboard` command in your Kickstart file is straightforward. The standard U.S. keyboard requires the following command:

```
keyboard us
```

The `keyboard` command in your `anaconda-ks.cfg` file should match your installation. But just in case, several dozen types are available, such as French (`fr`) and Spanish (`es`). A complete list is available in the Red Hat customization guide.

MOUSE

The `mouse` command in your Kickstart file represents your pointing device. It could be a touchpad or a tablet. For example, the following command represents a generic PS/2 mouse, connected to the standard PS/2 port (`psaux`):

```
mouse genericwheelps/2 --device psaux
```

If you want to configure a two-button mouse to emulate a third middle button, add the `--emulthree` switch to the end of this command. As described in Chapter 3, pressing the two mouse buttons together functions as a third button.

There are other mouse types, such as a standard USB mouse (`genericusb`), a Microsoft mouse (`microsoft`), or a Logitech mouse (`logitech`). A complete list is available in the Red Hat System Administration Guide.

TIMEZONE

The `timezone` command may be later in the file. It's straightforward; it specifies the time zone associated with your computer. If Linux is the only operating system you're installing, you should set the hardware clock to Greenwich Mean Time (`--utc`) which allows Linux to handle changes for daylight saving time. A typical `timezone` command looks like this:

```
timezone --utc America/New_York
```

NOTE *UTC stands for Universal Coordinated Time, which satisfies those who don't want to refer to the city of Greenwich in the United Kingdom.*

Another common way to specify a time zone is relative to GMT. For example, the following command specifies a time zone 14 hours *ahead* of GMT, which corresponds to Hawaii Standard Time:

```
timezone Etc/GMT-14
```

BOOTLOADER

You need a bootloader such as GRUB or LILO to start Red Hat Linux. The following `bootloader` command specifies the location, along with other kernel parameters that may be required:

```
bootloader --location=mbr --append hdc=ide-scsi
```

This command tells Kickstart to install your bootloader on the Master Boot Record (`mbr`). It also sends a configuration message to the kernel for a CD-writer. It allows Linux to make the secondary master IDE drive (`hdc`) look like a SCSI drive (`ide-scsi`).

NOTE *For more commands that you can* `--append` *to the kernel, run the* `man bootparam` *command.*

Graphics

The graphics command in a Kickstart file, `xconfig`, can appear complex. It's easier than it looks. Since you don't have to configure an X Window system in Red Hat Enterprise Linux, the `xconfig` command isn't required.

Let's analyze the `xconfig` command from my Kickstart file:

```
xconfig --card "VESA driver(generic)" --videoram 65536
➥--hsync 31.5-48.5 --vsync 40-70 --resolution 800x600
➥ --depth 24 --defaultdesktop gnome
```

This specifies a generic video card, a VESA driver (which is the new term associated with SVGA). If your other computers also have the same card and monitor, you should be able to keep these settings for your Kickstart file. However, in case you need to make changes, we've listed some `xconfig` settings in Table 5.4.

TABLE 5.4: KICKSTART *XCONFIG* SETTINGS

SETTING	DESCRIPTION
`--card` "*name*"	Specifies the make and model of the video card
`--videoram` *amount*	Notes the amount of video RAM
`--hsync` *range*	Lists the range for horizontal frequency, in KHz
`--vsync` *range*	Lists the range for vertical synchronization, in MHz
`--resolution` *horxvert*	Specifies the resolution on the monitor
`--depth` *num*	Notes the number of colors per pixel
`--defaultdesktop gnome`	Sets up GNOME as the default GUI desktop
`--startxonboot`	Starts the X Window when installation is complete
`--noprobe`	Specifies that the installation process shouldn't probe the monitor
`--monitor` *name*	Specifies the make and model of the monitor

TIP *If you don't want to configure the X Window with this Kickstart file, add the* `skipx` *command. If you don't configure the X Window or specify* `skipx`*, Anaconda stops the installation process to let you configure the X Window.*

Network Settings

In this section, we assume that you have one or more network cards in your computers. But in most cases, the Kickstart process uses Red Hat installation files from a remote computer on a network. Therefore, you'll need a command similar to the following to configure a network card on your computer:

```
network --device eth0 --bootproto dhcp
```

This command assumes you have an Ethernet network card and a DHCP server on your local network. If the DHCP server is on a remote network, you'll need to use the BOOTP protocol; just replace `dhcp` with `bootp` in the previous command. For more information on Ethernet, see Chapter 15; for more information on DHCP servers and BOOTP, see Chapter 19.

Alternatively, you could specify static IP address information. As you'll recall from Chapter 3, that includes an IP address (`--ip`), network mask (`--netmask`), gateway address (`--gateway`), and the IP address of a DNS server (`--nameserver`). You can also specify the hostname (`--hostname`) for this computer with the following command:

```
network --device eth0 --bootproto static --ip 192.168.12.20 --netmask
➥ 255.255.255.0 --gateway 192.168.12.11 --nameserver 207.217.126.81
➥ --hostname Enterprise3
```

NOTE *The* `network` *command in a Kickstart file must be on one line.*

The Root Password

Every Red Hat Enterprise Linux installation requires you to set a root password. This is a simple command, which can be configured in one of two ways:

```
rootpw Big747Ap
rootpw --iscrypted $1$ZIvDlQpJ$ptS2UJkTRngOTacYN22vR1
```

The first method includes the password in clear text, which is acceptable if you're using a local Kickstart file. However, it's possible to use a remote Kickstart file; in that case, it's best to encrypt the password, as we've done in the second example.

Firewalls

You can configure a firewall in the Kickstart file. As you've seen during the installation process, you can choose to activate a firewall (or not):

```
firewall --enabled
firewall --disabled
```

Assuming you want a standard firewall, you can customize it. For example, if you have two network cards, `eth0` and `eth1`, you may want to disable the firewall on one of the cards with the following command:

```
firewall --enabled --trust=eth1
```

There are several standard services that you can let through your firewall, including Secure Shell connections (`--port=ssh:tcp`), Telnet connections (`--port=telnet:tcp`), incoming e-mail (`--port=smtp:tcp`), incoming requests for web pages (`--port=http:tcp`), and incoming connections to an FTP server (`--port=ftp:tcp`).

You can let other services through the firewall, as long as you know the port number and associated protocol. For example, the following command sets up a high-security firewall that allows outside requests for regular and secure web pages:

```
firewall --enabled --http --port=443:tcp --port=443:udp
```

The numbers are TCP/IP ports that are defined in `/etc/services`, as described in Chapter 15.

Authentication Options

Authentication involves checking the credentials of a user. Normally, this means just the username and password. However, you can configure this process in a number of ways. The standard Kickstart configuration file sets up shadow passwords with MD5 encryption:

```
authconfig --enableshadow --enablemd5
```

Several authentication options are available. For example, you can set up NIS support (`--enablenis`), specify the NIS domain name (`--nisdomain` *name*) or the NIS server (`--nisserver` *name*), allow Kerberos passwords (`--enablekrb5`), and check passwords on a Samba or Microsoft Windows server (`--enablesmbauth`). An extensive array of additional options are available; see the Red Hat Enterprise Linux System Administration Guide for details.

Hard Drive Partition Setup

When Anaconda writes your configuration to `anaconda-ks.cfg`, the hard drive settings are disabled by default. If you're satisfied with the following commands, delete the hash marks (`#`) to activate them:

```
#clearpart --all --drives=sda,sdb,sdc
#part /boot --fstype ext3 --size=100 --ondisk=sda
#part / --fstype ext3 --size=10000 --grow --ondisk=sda
#part swap --size=256 --grow --maxsize=512 --ondisk=sda
```

The first command (`clearpart`) deletes all data from any existing Linux-formatted partitions (`--linux`) on the first SCSI hard drive (`sda`). A standard Enterprise server installation deletes all data from all formatted partitions (`--all`).

The next command sets up a partition (`part`) for the `/boot` directory. It's to be formatted (`--fstype`) to the ext3 filesystem, with a size of 100MB, on the first SCSI hard drive (`--ondisk=sda`).

The next command configures the root (`/`) directory with a size of at least 10GB on the first SCSI hard drive. However, the growable flag (`--grow`) is set, which allows the partition to fill available space on the first SCSI hard drive.

The next command in this set configures the swap partition, with a standard size of at least 256MB and a maximum size (`--maxsize`) of 512MB on the first SCSI hard drive. More extensive hard drive configurations are possible. For example, the following commands configure separate partitions for the `/boot`, `/usr`, `/home`, root (`/`), and `/var` directories, as well as a swap partition:

```
#clearpart --linux
#part /boot --fstype ext3 --noformat --onpart hda2
#part /usr --fstype ext3 --size=5500
#part /home --fstype ext3 --size=5000
#part / --fstype ext3 --size=1000
#part /var --fstype ext3 --size=5000
#part swap --size=512
```

The --noformat --onpart hda2 command switches use an existing partition, without reformatting it. Furthermore, the following commands configure six partitions usable by RAID arrays and three physical volumes suitable for Logical Volume Management (LVM):

```
#part raid.20 --size=100
#part raid.18 --size=100
#part raid.16 --size=100
#part raid.14 --size=100
#part raid.12 --size=100
#part raid.10 --size=100
#part pv.9 --size=100
#part pv.8 --size=100
#part pv.7 --size=100
```

Finally, the following raid command sets up the /home/mj directory on a RAID5 array of three partitions (with one spare). The volgroup and logvol commands configure an LVM group for the /home/ez directory:

```
#raid /home/mj --fstype ext3 --level=RAID1 --spares=1 raid.10 raid.16 raid.20
#raid /home/dl --fstype ext3 --level=RAID5 raid.12 raid.14 raid.18
#volgroup Volume00 pv.7 pv.8 pv.9
#logvol /home/ez --fstype ext3 --name=LogVol00 --vgname=Volume00         --size=280
```

One other simple command ensures that the system reboots after the Kickstart installation process is complete:

```
reboot
```

TIP *Don't forget to remove the boot media after installation starts; otherwise, your users may see the first installation step when they get to their computers in the morning.*

Packages and Groups

When you see the %packages command, the items that follow specify the packages and groups that will be installed. The first lines in this section should look similar to the following, which specifies five package groups. If you review the comps.xml file, you'll recognize these as the <id> variable associated with different package groups.

```
%packages
@ office
@ mysql
@ system-tools
@ base-x
@ graphics
```

These commands search through the comps.xml file described in the first part of this chapter for groups with the given names, per the <name> variable in the comps.xml file. Some of the packages in each group—as indicated by <packagereq type="mandatory">—must be installed. Other packages may be "default" or "optional". You may have selected or deselected some of these packages during

the Red Hat Enterprise Linux installation process. This is followed by the name of two key packages, which may not be part of any specific package group. The following makes sure that a Kernel and bootloader are installed:

```
kernel
grub
```

Postinstallation Commands

Once Linux is installed, Kickstart proceeds to the postinstallation script at the end of the file. You can run the full range of available scripts; the default language is based on the bash shell. To specify a different scripting language, use a command such as the following:

```
%post --interpreter /usr/bin/python
```

You can copy more configuration files from a remote computer; for example, the following script copies the `XF86Config` file from the computer with the noted IP address:

```
mkdir /mnt/source
mount 192.168.0.1:/etc /mnt/source
cp /mnt/source/etc/X11/XF86Config /root
```

This assumes you've shared the `/etc` directory via NFS on the computer with the noted IP address.

Other Commands

A substantial number of commands are available for Kickstart files. Table 5.5 lists many of the basic Kickstart commands.

TABLE 5.5: OTHER KICKSTART COMMANDS

COMMAND	DESCRIPTION
`autopart`	Configures a default set of partitions, including a root directory (`/`) greater than 1GB, `/boot`, and swap.
`auth`	Lets you specify authentication options; same as `authconfig`. Many authentication options available.
`bootloader`	Specifies the bootloader location; `--useLilo` installs LILO instead of GRUB; `--password=`*`password`* sets a GRUB loader password.
`clearpart`	Removes current partitions; you can specify `--linux` or `--all`.
`device`	Allows you to set hardware parameters for a specific device.
`driverdisk`	If you need a separate driver disk, you can load it onto an existing partition or even a network source; for example, you can use `driverdisk hda2 --type=vfat` or `driverdisk --source =ftp://`*`drvdisk`*`.img`
`firewall`	Lets you set up a basic firewall configuration.

Continued on next page

Table 5.5: Other Kickstart Commands *(continued)*

Command	Description
`install`	Allows you to specify basic installation parameters, including the source of Red Hat installation files.
`interactive`	Runs through the Kickstart file interactively; same as `autostep`.
`keyboard`	Specifies the keyboard type.
`lang`	Notes the language of the installation; somewhat irrelevant for an automated installation.
`langsupport`	Specifies the language(s) you want to install.
`logvol`	Adds a logical volume partition.
`mouse`	Adds a pointing device.
`network`	Configures the local network card.
`part`	Creates a specified partition; same as `partition`.
`raid`	Configures a software RAID device.
`reboot`	Reboots the system after the installation is complete.
`rootpw`	Specifies the root password for this system.
`skipx`	Skips the X Window configuration process.
`text`	Runs the installation in text mode; somewhat irrelevant for an automated installation.
`timezone`	Specifies the time zone for this computer.
`upgrade`	Upgrades an existing Linux system.
`volgroup`	Sets up an LVM group.
`xconfig`	Notes X Window and graphics card configuration details.
`zerombr`	Overwrites any existing partition tables, including all bootloaders.

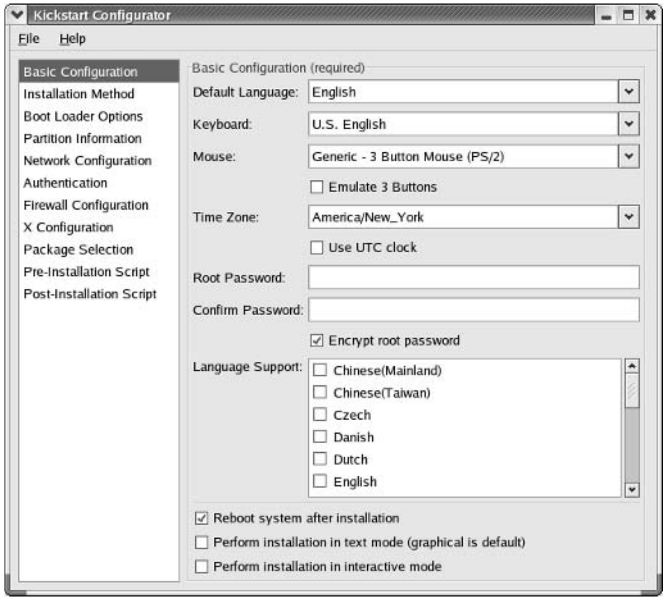
NOTE *Many of these commands have a wide variety of switches. We've covered the ones we consider to be important in this chapter. If you need more information, refer to the Red Hat Enterprise Linux 3 System Administration Guide.*

Working with the GUI Kickstart Configurator

There's another way to create a custom Kickstart configuration file: using the GUI Kickstart Configurator. You can start it in a GUI such as GNOME or KDE. Open a command-line interface and run the `redhat-config-kickstart` command to open the Kickstart Configurator, shown in Figure 5.3.

NOTE *If you need more information on starting a command-line interface in GNOME or KDE, refer to Chapter 30.*

FIGURE 5.3
The Kickstart Configurator



As you can see, the left-hand column contains 11 menus, which we'll look at in the following sections. If you've installed Red Hat Enterprise Linux or read the first parts of this chapter, you should already be familiar with many of the options.

If you want to start from an existing configuration, select File ➢ Open File. You can then select a file, such as `/root/anaconda-ks.cfg`, from the menu that appears. You can then start with some of the defaults for when you installed Linux on the local computer. In my experience, this tool is less than perfect; you may need to modify a few settings before using the resulting Kickstart configuration file.

The Basic Configuration Menu

The Basic Configuration menu is shown in Figure 5.3. It includes a number of basic settings, which are briefly described in Table 5.6.

TABLE 5.6: KICKSTART CONFIGURATOR, BASIC CONFIGURATION OPTIONS

Option	Description
Default Language	Specifies the language you want to use during the installation process; 19 languages are available.
Keyboard	Specifies a keyboard type; you can select from more than 50 keyboards.
Mouse	Selects the mouse or other pointing device for your computer.
Emulate 3 Buttons	If you have a two-button mouse, this option allows you to simulate a middle mouse button by pressing both mouse buttons at the same time.

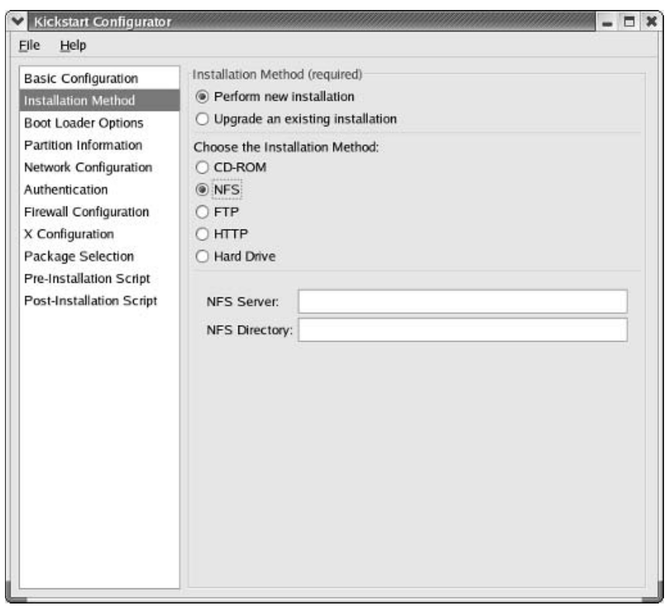
Table 5.6: Kickstart Configurator, Basic Configuration Options *(continued)*

Option	Description
Time Zone	Specifies your current time zone.
Use UTC Clock	Select this option if you've set your hardware clock to Greenwich Mean Time and are not dual-booting with an operating system such as Microsoft Windows.
Root Password	Enter your desired root password here.
Encrypt Root Password	Encrypts the root password that you enter in the Kickstart file.
Language Support	Installs fonts and language files for your running Linux computer.
Reboot System After Installation	Adds the `reboot` command to the Kickstart file.
Perform Installation In Text Mode	Runs the installation process in text mode.
Perform Installation In Interactive Mode	Allows you to debug a Kickstart installation process.

The Installation Method Menu

In the Kickstart Configurator, select Installation Method. You should see the options shown in Figure 5.4.

Figure 5.4
The Kickstart Configurator's Installation Method menu

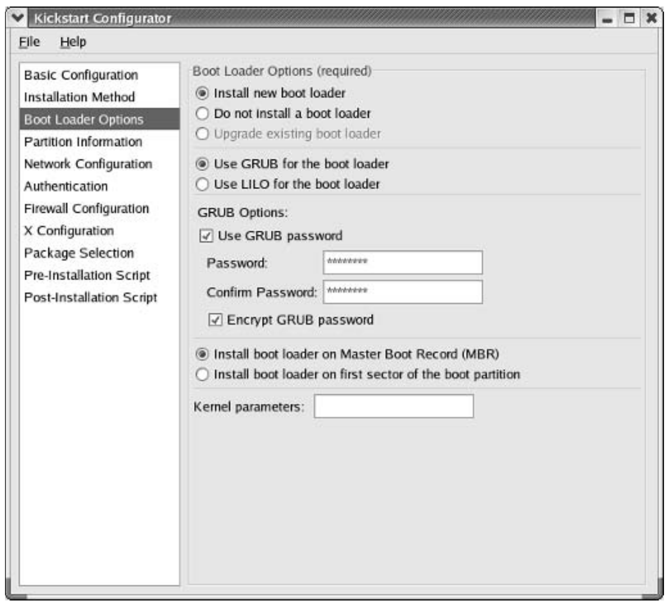


These options are fairly self-explanatory; you can configure Kickstart to install a fresh copy or upgrade Red Hat Enterprise Linux. You can also specify a local (CD-ROM or Hard Drive) or network (FTP, HTTP, NFS) source for the installation files. When you do, additional options appear so you can specify where the installation files are located.

The Boot Loader Options Menu

The Boot Loader Options menu allows you to configure the type and location of the bootloader on your system. As we discussed in Chapter 11, there are two basic Linux bootloaders: GRUB and LILO. As you can see in Figure 5.5, this menu contains five sections.

Figure 5.5
The Boot Loader Options menu



If you already have a third-party bootloader (from Partition Magic or System Commander, for example), you can install GRUB or LILO on the first sector of the boot partition.

You can select GRUB or LILO as your bootloader. If you select LILO, you'll see slightly different options. You can have it read your hard disks in `linear` mode, which can help with larger hard drives. You can also force the use of `lba32` mode, which can help Linux look beyond the 1,024th cylinder on older hard drives for the startup files in your `/boot` directory.

Normally, you'll install the bootloader on the Master Boot Record. If you prefer to use another bootloader, you can install GRUB or LILO on the first sector of the partition with your `/boot` directory.

You can also pass hardware parameters to the kernel. This is most commonly used when Linux has trouble detecting hardware automatically. You can specify a wide variety of parameters here, as defined in the `bootparam` man page.

NOTE *In Linux, a* man page *is a manual, typically for commands or configuration files. For example, to read the man page for* `/etc/fstab`*, open a Linux command-line interface and run the* `man fstab` *command.*

There is one more option related to bootloaders, which we discuss in the next section.

The Partition Information Menu

You can configure most of the partitions you need in the Partition Information menu, shown in Figure 5.6.

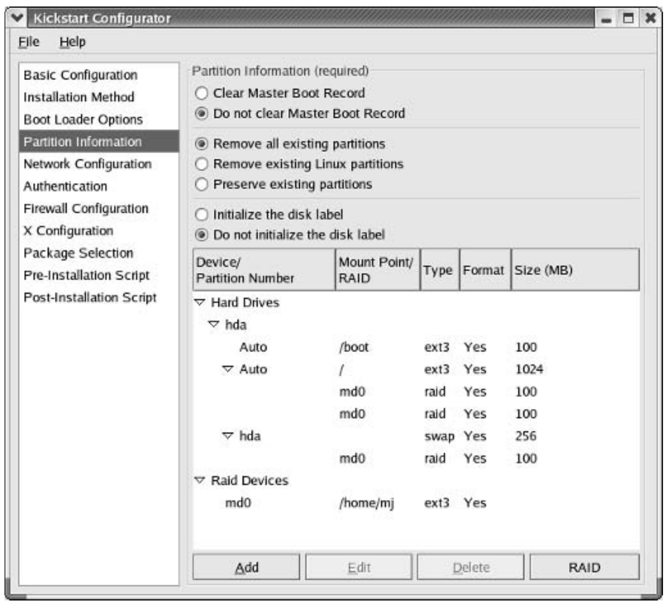


FIGURE 5.6
The Partition Information menu

The first parts of this menu allow you to set basic parameters for your hard disk. The Clear Master Boot Record option erases any existing bootloader from your hard disk. It's equivalent to Kickstart's `zerombr=yes` command.

If the hard disks have existing partitions, you can choose to delete just the Linux partitions, or all partitions on all detected hard drives. If you're installing Linux on computers with new hard drives, you'll also want to select Initialize The Disk Label.

NOTE *If you're upgrading Red Hat Enterprise Linux, you're normally using existing partitions; all of the options in this menu are then deactivated.*

Click Add to open the Partition Options dialog box, shown in Figure 5.7. If you're familiar with Disk Druid from Chapter 3, the options here should look familiar. If you need more information on most of these options, read Chapter 3.

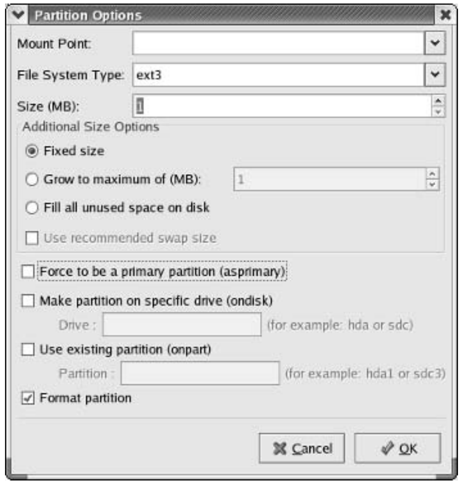


FIGURE 5.7 The Partition Options dialog box

In addition to what is shown in Disk Druid, this dialog box contains the following two options:

Use Recommended Swap Size Red Hat can configure a recommended swap partition. It's normally twice the size of your RAM.

Use Existing Partition If you know the partition layout of the target computer, you can specify a partition such as `hda1`. See Chapter 2 for partition-naming conventions.

As of this writing, the Kickstart Configurator does not support the format of volume groups for LVM partitions. You can still add LVM criteria to the actual Kickstart file, as we explained earlier.

You can also set up RAID devices. If you've configured RAID partitions, click RAID. In the RAID Options window, select Create A RAID Device and click OK to continue. This opens the Make RAID Device dialog box, shown in Figure 5.8.

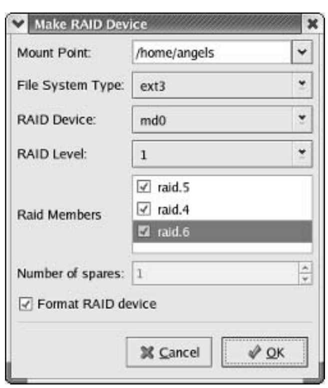


FIGURE 5.8 The Make RAID Device dialog box

If you have a sufficient number of RAID partitions, this dialog box supports creating RAID devices at levels 0, 1, and 5. For more information on RAID requirements at these levels, see Chapter 14.

The Network Configuration Menu

To configure Ethernet network cards on your computer, use the Network Configuration menu. If you have a different type of network card, you'll have to edit the Kickstart configuration file directly. As you can see in Figure 5.9, the buttons allow you to add, edit, or delete various network devices.

When you add or edit a network device, it opens the Network Device Information dialog box, also shown in Figure 5.9.

You can configure a number of settings for each network device:

Network Device Click the drop-down arrow to set this to one of 17 Ethernet network devices, between eth0 and eth16.

Network Type You can select a network type for Static IP configuration; or you can get data for this network device from a local DHCP server or a remote DHCP server using BOOTP. If you choose to set a Static IP network type, you can configure network address information for that device.

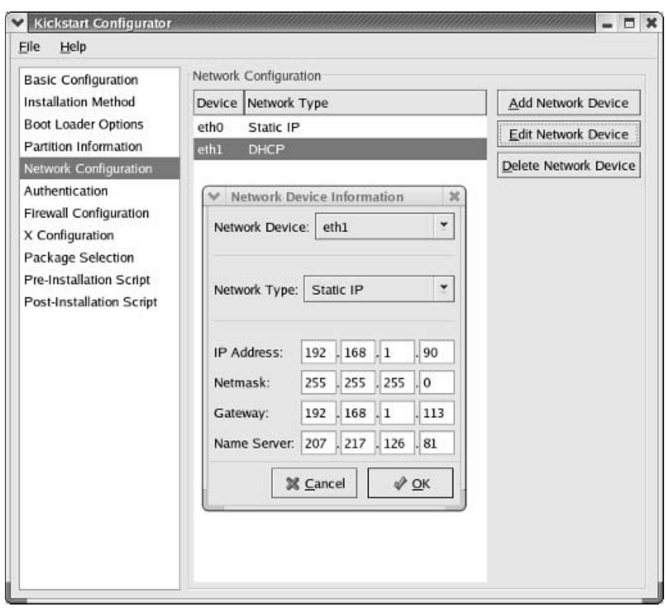
IP Address The IP version 4 address for the network card

Netmask The network mask for your LAN

Gateway The IP address of the computer or router that connects your network to an external network such as the Internet

Name Server The IP address of a DNS server connected to your network

FIGURE 5.9 The Network Configuration menu



If you're unfamiliar with the basics of IP addressing, more information on each of these settings is available in Chapter 15.

The Authentication Configuration Menu

As we described earlier, authentication normally describes how a computer checks usernames and passwords. The basic menu is shown in Figure 5.10.

By default, Kickstart configures two types of password security. Shadow passwords are part of the Shadow Password Suite described in Chapter 9. MD5 is a form of encryption applied to user passwords.

As you can see in Figure 5.10, this window includes a series of tabs that represent various forms of authentication. They are briefly described in Table 5.7.

FIGURE 5.10
The Authentication Configuration menu

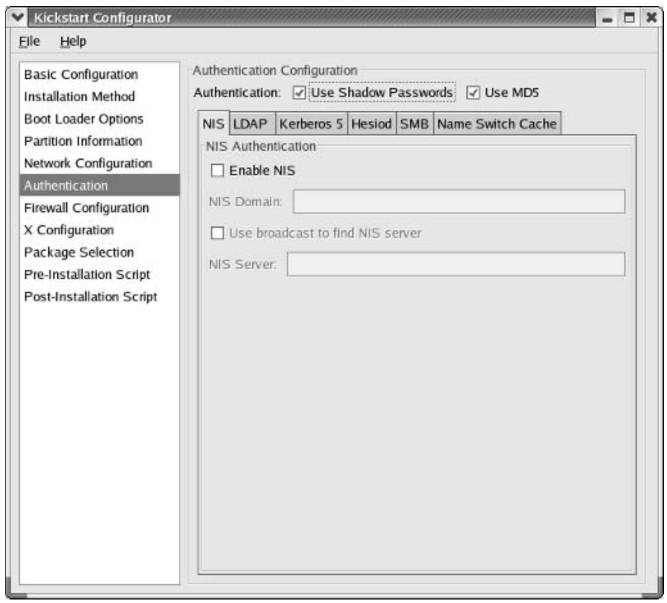


TABLE 5.7: THE KICKSTART CONFIGURATOR AUTHENTICATION OPTIONS

OPTION	DESCRIPTION
NIS	Network Information Service provides a common database of usernames and passwords for a LAN; for more information, see Chapter 23.
LDAP	The Lightweight Directory Assistance Protocol is also used for authentication and related LAN databases; for more information, see Chapter 23.
Kerberos 5	Developed at MIT, Kerberos 5 provides strong encryption for checking user credentials.

Continued on next page

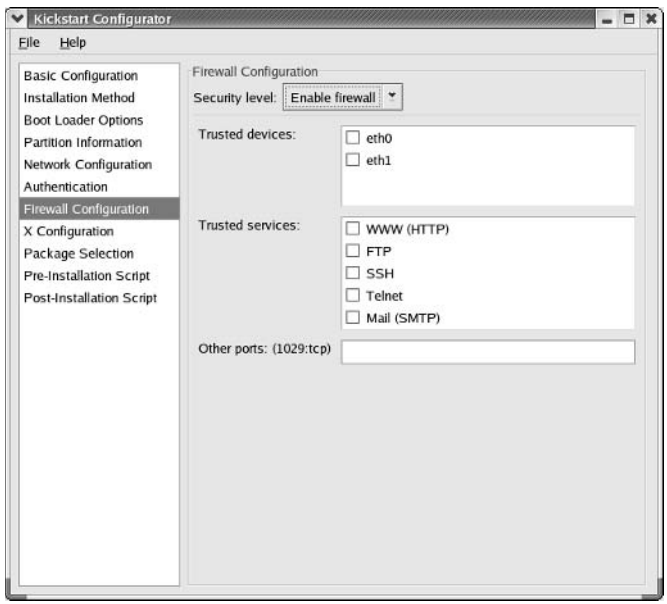
TABLE 5.7: THE KICKSTART CONFIGURATOR AUTHENTICATION OPTIONS *(continued)*

Option	Description
Hesiod	Functionally similar to NIS, `hesiod` uses DNS to distribute information kept in basic configuration files.
SMB	The SMB (Samba) option allows you to use other servers for authentication on a Microsoft Windows–based network.
Name Switch Cache	The associated daemon, `ncsd`, supports authentication via NIS.

The Firewall Configuration Menu

The Firewall Configuration menu should look familiar if you've installed Red Hat Enterprise Linux in either Chapter 3 or 4. As you can see in Figure 5.11, you can select Enable Firewall or Disable Firewall.

FIGURE 5.11
The Firewall Configuration menu



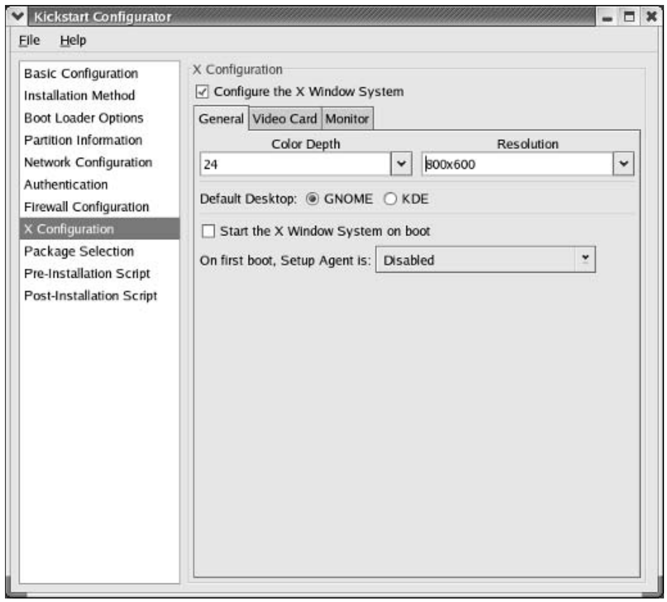
If you choose to enable a firewall, you can customize it. You can exclude a network card such as `eth0` from the firewall by checking the device name in the Trusted Devices text box. In addition, you can allow incoming network traffic to several different types of servers: web (WWW), FTP, a Secure Shell (SSH), Telnet, and incoming mail (SMTP).

The Other Ports text box lets you add other ports based on `/etc/services`.

The X Configuration Menu

The X Configuration menu should look familiar if you know about the `redhat-config-xfree86` tool. If you choose to configure the X Window through Kickstart, select Configure The X Window System. This activates the three tabs shown in Figure 5.12.

FIGURE 5.12
The X Configuration menu



On the General tab, you can select an overall color depth and resolution for your system. Be careful; some systems can handle a color depth of 24 bits per pixel, and others are designed for 32 bits per pixel. Assuming your computer reflects the target hardware, it's best to take a working configuration from the `xconfig` command in your `anaconda-ks.cfg` file.

If you've installed GNOME and/or KDE, you can designate either of these as your default desktop. If you enable the Start The X Window System On Boot option, Linux opens one of the display managers described in Chapter 29. After Linux boots the first time, you can Disable or Enable the Red Hat Setup Agent. Also known as `firstboot`, we covered this process in Chapter 3. The drawback is that you cannot use Kickstart to automate responses to the Setup Agent, and it therefore can block an automated installation.

The Video Card and Monitor tabs include the same database that is available through `redhat-config-xfree86`. You can find more information on this system in Chapter 29. By default, Kickstart probes for your video card and monitor, or you can activate the settings, including the monitor horizontal and vertical sync, using this tool.

The Package Selection Menu

The Package Selection menu allows you to select from the standard package groups in the `comps.xml` configuration file. As shown in Figure 5.13, the window is organized in the same way as Red Hat's graphical installation tool.

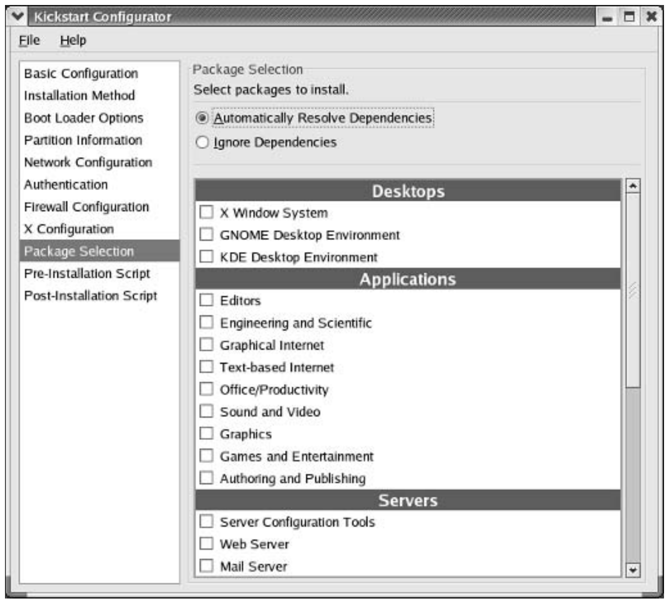


FIGURE 5.13 The Package Selection menu

Select the package groups of your choice. Details of each group are available in the `comps.xml` file. However, the current version of the Kickstart Configurator does not allow you to select several package groups, including those related to SQL databases and legacy software.

Unless you know what you're doing, select Automatically Resolve Dependencies. That option ensures that foundation software gets installed. Otherwise, a lot of the software installed with Red Hat Enterprise Linux may not work.

The Pre-Installation Script Menu

As we explained earlier, a preinstallation script helps you set parameters for the installation. Since the script is run before Red Hat Enterprise Linux is installed, the range of available commands is limited. You can use the Kickstart Configurator to create a preinstallation script.

The default script language is bash. If you want to use commands in a different language, activate the Use An Interpreter text box and then enter the location of another language module, such as `/usr/bin/python`. Test your scripts; if there's an error, your Kickstart installation may fail.

The Post-Installation Script Menu

A postinstallation script helps you add parameters for each configuration. You can also use the Kickstart Configurator to create a postinstallation script.

Postinstallation scripts are run in a `chroot` environment. In other words, during the installation process, the standard Linux root directory is mounted on the `/mnt/sysimage` directory. The following command makes `/mnt/sysimage` look like your root directory:

```
# chroot /mnt/sysimage
```

Once again, it's important to test your scripts. If there's an error, your Kickstart installation may fail.

The Next Steps

Once you've made your changes, you'll want to save your configuration to a Kickstart file. To do so, select File ➢ Save File and save the file in the directory of your choice. As you'll see in the next section, it helps to name the file `ks.cfg`.

If there are things you could not add to your configuration file, such as LVM partitions, open `ks.cfg` in a text editor and do so now. We examined the basic configuration and commands of a Kickstart file earlier in this chapter.

Kickstarting from a Boot Disk

Now that you have a Kickstart file, you should be able to start the Red Hat Enterprise Linux installation process from the installation boot floppy or CD. Once Red Hat finds and loads your Kickstart file, it may need a driver disk. After it activates needed drivers, Anaconda proceeds to install Red Hat Enterprise Linux automatically, using the instructions from your Kickstart file. You can then remove the installation and driver disks and use them to start the process on another computer.

In other words, you can install Red Hat Enterprise Linux on several computers simultaneously.

Files on a Boot Floppy

Kickstart files are typically small enough to include with the standard Red Hat Enterprise Linux installation floppy disk. The standard files from the `bootdisk.img` boot floppy are shown in Figure 5.14. Note that I've included my `ks.cfg` file on this floppy.

Create a Red Hat Enterprise Linux installation floppy, using the techniques described in Chapter 3. Rename any Kickstart file you've created as `ks.cfg`. Copy this file to the installation floppy.

Unfortunately, there isn't enough room to include drivers on the installation boot floppy. If you need additional drivers, you can use the installation boot floppy and the Red Hat Enterprise Linux installation CD.

NOTE *Some companies buy PCs without CD drives in an attempt to prevent users from loading their own software.*

If your computer does not have a CD drive, you'll need more floppy disks for any drivers that Linux needs to load. As described in Chapter 2, other floppies can be created from the first Red Hat installation CD, from files in the `/images` directory. Depending on your configuration, you may need floppies created from one or more of the following: `drvnet.img`, `drvblock.img`, and `pcmciadd.img`.

FIGURE 5.14
Files on the installation boot floppy

```
[root@Enterprise3 root]# \ls -l /mnt/floppy/
total 1412
-rwxr-xr-x    1 root     root          364 Oct  7 19:26 boot.msg
-rwxr-xr-x    1 root     root         1026 Oct  7 19:26 general.msg
-rwxr-xr-x    1 root     root       534021 Oct  7 19:26 initrd.img
-rwxr-xr-x    1 root     root         1528 Feb 25 14:55 ks.cfg
-r-xr-xr-x    1 root     root         7856 Oct  7 19:26 ldlinux.sys
-rwxr-xr-x    1 root     root          660 Oct  7 19:26 options.msg
-rwxr-xr-x    1 root     root          869 Oct  7 19:26 param.msg
-rwxr-xr-x    1 root     root          557 Oct  7 19:26 rescue.msg
-rwxr-xr-x    1 root     root          549 Oct  7 19:26 snake.msg
-rwxr-xr-x    1 root     root         6056 Oct  7 19:26 splash.lss
-r-xr-xr-x    1 root     root          435 Oct  7 19:26 syslinux.cfg
-rwxr-xr-x    1 root     root       888750 Oct  7 19:26 vmlinuz
[root@Enterprise3 root]#
```

NOTE *If you have computers with bootable network cards, you can also use the PXE Boot Server described in Chapter 4*

Files on a Boot CD

One solution to the driver problem is to configure the Kickstart file on a boot CD. You can set it up based on the files embedded in the `boot.iso` file. In Chapter 3, we described how this file can be configured as a boot CD. Now, we'll show you how you can add the Kickstart configuration file to this CD. You can do this with the following steps. While they are slightly different from what you'll find in the Red Hat System Administration Guide, they work for me

1. Mount the first Red Hat installation CD on the `/mnt/cdrom` directory.

   ```
   # mount /mnt/cdrom
   ```

2. Find the `boot.iso` file in the `images/` subdirectory, and mount it as a loop device on an empty directory. We have created `/mnt/source` available for this purpose and explain this command in Chapter 14:

   ```
   # mount -t iso9660 -o loop /mnt/cdrom/images/boot.iso /mnt/source
   ```

3. Now you can view individual files in the `/mnt/source` directory and copy them to a writeable directory such as `/tmp/boot`.

   ```
   # cp -ar /mnt/source/* /tmp/boot
   ```

4. Copy your new Kickstart configuration file to the `isolinux/` subdirectory.

   ```
   # cp ks.cfg /tmp/boot/isolinux
   ```

5. Make sure the permissions support booting.

```
# chmod u+w /tmp/boot/isolinux/*
```

6. Now the following command is a little complex. It allows you to create a new boot image (`boot1.iso`), using the `isolinux.bin` boot image, with the `boot.cat` boot catalog, and it configures the image on a nonfloppy (`-no-emul-boot`), with standard sectors (`-boot-load-size`) and a standard boot table (`-boot-info-table`). Do not overlook the last dot in the command, which represents the current directory.

```
# cd /tmp/boot/isolinux
# mkisofs -o /tmp/boot1.iso -b isolinux.bin -c boot.cat -no-emul-boot
-boot-load-size 4 -boot-info-table -R -J -v -T .
```

7. Finally, write your new boot image to a CD.

```
# cdrecord -v speed=2 dev=0,0,0 /tmp/boot1.iso
```

We've used a specific set of commands, which we explain in more detail in Chapter 14. You can now use this boot CD to start the installation of Red Hat Enterprise Linux 3. When you see the first installation boot prompt, you enter the following command to get Kickstart to install Linux automatically:

```
boot: linux ks=cdrom:/ks.cfg
```

The Installation Procedure

You're ready with your installation disk. Insert the Red Hat Enterprise Linux installation disk with your Kickstart file into the appropriate drive. If possible, insert the first Red Hat Enterprise Linux installation CD. Restart your computer, and boot from the installation floppy or the CD. When you see the first installation screen, you'll see the boot prompt, where you can enter the following command to start a Kickstart installation from a boot floppy:

```
[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: linux ks=floppy
```

If you've configured your `ks.cfg` file properly and booted from the CD, you should be able to remove the floppy and the CD after your computer reads in the startup kernel and appropriate drivers. The installation should proceed automatically. Alternatively, if you've set things up on a boot CD, enter the following command:

```
boot: linux ks=cdrom:/ks.cfg
```

If you don't boot from a CD, you'll have to insert the appropriate driver floppy disks when prompted. The prompts will be similar to the driver screens shown in Chapter 4. You can even use a Kickstart file from a remote computer on a network. For example, if you've copied that file to an NFS server, you'd use the following command at the first installation boot prompt:

```
boot: linux ks=nfs:server.example.com:/mnt/inst/ks.cfg
```

Testing Kickstart

Kickstart is useful for installing Red Hat Enterprise Linux on a group of computers with similar or identical hardware configurations. If you're going to install Kickstart on a large number of computers, it's important to test your installation first.

If you're planning to install Red Hat Enterprise Linux on a large group of computers, you could stay in the office all night to make sure everything goes right, or you could test your Kickstart installation process on one or two computers. Then you can use Kickstart to install Red Hat Enterprise Linux (or one of the freely available third-party rebuilds that we've described in Chapter 1) on the other computers on your network with additional confidence.

Summary

In previous chapters, we found that the installation of Red Hat Enterprise Linux can be an involved process. Anaconda, the Red Hat installation program, can require considerable user input. In this chapter, you learned how to install Red Hat Enterprise Linux automatically, using Kickstart. With an appropriate Kickstart file, you can insert a floppy and a CD into a computer and then type a simple command, and the installation proceeds automatically.

To demonstrate how to configure a Kickstart file, we examined the `comps.xml` file, which organizes Red Hat Enterprise Linux packages into several groups.

Then we examined the default Kickstart configuration for a computer, which is saved in the `/root` directory in `anaconda-ks.cfg`. This file, with some modifications, allows Kickstart to create the same configuration on another computer.

The Kickstart Configurator provides a GUI interface for creating a custom Kickstart file. While creating a basic configuration saves you time, you may need to add a few more commands to the resulting file in a text editor.

Once you're satisfied with your Kickstart file, you can save it to `ks.cfg` on a Red Hat Enterprise Linux installation boot floppy, CD, or network server. You can use the first Red Hat Enterprise Linux installation CD or a driver floppy for required drivers. If `ks.cfg` is properly configured, a simple command starts the installation. Unless you need to insert a separate driver floppy, you should be able to walk away from the computer. Red Hat Enterprise Linux is installed automatically.

In the next chapter, we'll begin our journey through the nitty-gritty of Linux, the command-line interface. We'll examine the basic commands required to navigate around and administer Linux in the chapters that follow.

Part 2

Linux Fundamentals

In this Part, you will learn:

Chapter 6

Starting at the Command Line

While Red Hat Enterprise Linux includes a number of integrated GUI tools, the best way to control Linux is from the command-line interface. Command-line tools have more options than GUI tools. Since they don't include the overhead of a desktop such as GNOME or KDE, they are faster. And there is still a strong bias in the Linux community toward the command line. Therefore, if you really want to learn Linux, you should learn how to use the command-line interface.

This chapter shows you the workings of a number of different commands based on the Bourne Again Shell (bash), discussed in Chapter 8. Some commands help you navigate different Linux directories; others help you create and delete Linux files. Commands are available to help you read or search through files in different ways. Some commands allow you to use the characteristics of a file to your advantage.

One of the keys to the command-line interface is the `vi` editor, which may be the only editor you have available if you're troubleshooting problems such as boot failures. This chapter covers the following topics:

- Exploring navigational commands
- Setting up files and directories
- Managing files
- Manipulating files
- Using the vi editor
- Understanding other text editors

Exploring Navigational Commands

There are two basic navigational commands for getting around the shell. The `cd` command lets you navigate between directories. The `ls` command tells you the contents of a directory (including other directories). But before you run around different Linux directories, the `pwd` command can tell you where you are. Output from a navigational command depends on the absolute path, which specifies your directory location relative to the top-level root (`/`) directory.

pwd

The `pwd` command (while it's short for *print working directory*, it is unrelated to printers) is simple. Type it at the command-line interface, and you'll see the absolute path to your current directory. For example:

```
# pwd
/etc/httpd/conf
```

The output tells you that you're currently in the `/etc/httpd/conf` directory, which happens to be the default location for Apache configuration files.

cd

The change directory command is known as `cd`. Those of you familiar with MS-DOS may find a number of similarities between MS-DOS and Linux `cd` commands. Typical `cd` commands are shown in Table 6.1.

NOTE *Linux is case sensitive. Please note that the small capitals in the tables of this chapter represent lowercase letters.*

TABLE 6.1: *CD* COMMANDS

COMMAND	RESULT
`cd ..`	Moves up one directory level. For example, if you're currently in the `/home/mj` directory, this moves you to the `/home` directory.
`cd ../..`	Moves up two directory levels. For example, if you're currently in the `/etc/rc.d/rc0.d` directory, this moves you to the `/etc` directory. You can move up additional directory levels, up to the root (/) directory.
`cd /home/mj`	Navigates to the home directory of user mj.
`cd ~`	Navigates to your home directory. Works for any user.

NOTE *If you're relatively new to Linux, remember to use the forward slash* `/`*, not the backslash* *, when you cite directory, computer, or even domain names.*

ls

The `ls` command is versatile. Not only does it allow you to list the files and directories in your current directory, but with the proper options, you can also find the permissions and size of a file. The command allows you to check ownership, differentiate between file types, and sort the result in several ways. You can review some examples of this command in Table 6.2.

Perhaps the most important command in this series is `ls -l`, which lists all files in the current directory, including size, owner, and permissions. Figure 6.1 shows an example of the result of this command.

TABLE 6.2: *LS* COMMANDS

COMMAND	**RESULT**
`ls`	Lists in alphabetical order all nonhidden files in the current directory.
`ls -a`	Lists all files in the current directory, including hidden files.
`ls -r`	Lists in reverse alphabetical order all nonhidden files in the current directory.
`ls -F`	Lists all files by type. The character at the end of each file indicates the file type. For example, a forward slash (/) represents a directory, an asterisk (*) is associated with an executable file, and an at sign (@) represents a linked file.
`ls -i`	Lists files with inode numbers. An inode number represents the location of a file on a volume. Two or more files with the same inode number are two different names for the identical file.
`ls -l`	Lists all the files in the current directory, including the current directory (`.`) and the parent directory (`..`). Also lists the size, owner, and permissions associated with each file in what's known as *long listing format*.
`ls -t`	Lists files by the last time they were changed; most recent files are listed first.
`ls -u`	Lists files by the last time they were accessed; most recent files are listed first.

As you can see, the long listing includes the permissions, user owner, group owner, size, modification time, and name of each file in the current directory.

TIP *Normally, the output from commands such as* `ls` *are color coded; green output usually represents a directory. If you add a backslash in front of the command, that removes color from the output. For example, I used the* `\ls -l` *command to prepare Figure 6.1.*

FIGURE 6.1
A long listing (ls -l) in the current directory

```
-rw-------    1 root     root          125 Sep 15 14:38 vsftpd.ftpusers
-rw-------    1 root     root          361 Sep 15 14:38 vsftpd.user_list
-rw-r--r--    1 root     root         1305 Aug 16  2003 warnquota.conf
-rw-r--r--    1 root     root        23735 Aug  2  2003 webalizer.conf
-rw-r--r--    1 root     root        23930 Aug  2  2003 webalizer.conf.sample
-rw-r--r--    1 root     root         4022 Jun 25  2003 wgetrc
-rw-r--r--    1 root     root            0 Feb 26 22:10 wvdial.conf
drwxr-xr-x   16 root     root         4096 Feb 26 18:59 X11
-rw-r--r--    1 root     root          289 Sep  2 22:20 xinetd.conf
drwxr-xr-x    2 root     root         4096 Dec 10 17:39 xinetd.d
drwxr-xr-x    2 root     root         4096 Oct 23 15:36 xml
-rw-r--r--    1 root     root         4914 Jul 29  2003 xpdfrc
-rw-r--r--    1 root     root         5475 Jul 29  2003 xpdfrc.ja
-rw-r--r--    1 root     root         5334 Jul 29  2003 xpdfrc.ko
-rw-r--r--    1 root     root         5536 Jul 29  2003 xpdfrc.zh_CN
-rw-r--r--    1 root     root         5623 Jul 29  2003 xpdfrc.zh_TW
-rw-r--r--    1 root     root          501 Oct 23 16:20 yp.conf
-rw-r--r--    1 root     root         1626 Jun  3  2003 ypserv.conf
-rw-r--r--    1 root     root          253 Oct 14  2002 zlogin
-rw-r--r--    1 root     root           86 Oct 14  2002 zlogout
-rw-r--r--    1 root     root          146 Oct 14  2002 zprofile
-rw-r--r--    1 root     root          304 Nov 28  2002 zshenv
-rw-r--r--    1 root     root          627 May  1  2003 zshrc
[root@Enterprise3 etc]#
```

Path Management

When you describe the location of a file, you specify either the *absolute* path or the *relative* path. An absolute path describes the location of a file relative to the root (`/`) directory. For example, you can type the following command to get to the scripts that start a number of Linux daemons:

```
# cd /etc/rc.d/init.d
```

The forward slash in front of the first directory makes this the absolute path. You can type this command from anywhere in Linux to get to this directory. Sometimes, you may accidentally type the command without the forward slash:

```
# cd etc/rc.d/init.d
```

If you've just logged in, Linux looks for these directories under your home directory. For example, if your home directory is `/home/mj`, this command makes Linux look for the `/home/mj/etc/rc.d/init.d` directory. Unless you keep a copy of these files deep in your home directory, Linux won't find anything.

Absolute and relative paths apply to other commands as well. For example, you can list the daemons in the `/etc/rc.d/init.d` directory with the following command:

```
# ls /etc/rc.d/init.d
```

However, if you use the relative path, your current directory matters. For example, if the output from the `pwd` command is `/home/mj`, the following command won't work unless you have a `/home/mj/etc/rc.d/init.d` directory:

```
# ls etc/rc.d/init.d
```

Setting Up Files and Directories

Creating a file in Linux is easy. You can copy from an existing file or save to the filename of your choice from an editor or another application. There's even a special command that allows you to set up an empty file. It's also easy to delete a file—so easy that some commands for deleting files can be dangerous.

Although a Linux directory is just a special file, Linux includes specific commands for creating and deleting directories. First, we'll look at the file management commands, and then we'll examine the commands for creating and deleting directories.

touch

There are times when you simply need to set up an empty file in Linux. For example, before you can activate a quota for a user or a group, you need to create an empty `aquota.user` or `aquota.group` file in the target directory. Creating empty files is easy with the `touch` command. The following commands create these files in the `/home` directory:

```
# touch /home/aquota.user /home/aquota.group
```

The `touch` command can also be used to change the timestamp associated with an existing file. When you use the command without a switch, the last access time of the file is changed to the current time. For example, suppose it's 11:21 on April 15 and you run the following command:

```
# touch /root/f0601.tif
```

When you run the `ls -l` command on the `f0601.tif` file, you see the following output:

```
-rw-r--r--   1 root root   883823 Apr 15 11:21 f0601.tif
```

Other switches, such as `-t`, can change the access time associated with a file as desired.

cp

The simplest version of the copy command is `cp` *`file1`* `file2`. Issuing this command copies the contents of *`file1`* and places them in destination *`file2`*. The destination file will have a new creation date and inode number. Other copy commands can overwrite destination files. You can even use a switch for the `cp` command to copy the contents of one or more subdirectories. See Table 6.3 for examples of how the `cp` command works.

TABLE 6.3: *CP* COMMANDS

COMMAND	RESULT
`cp` *`file1 file2`*	Copies the contents of source `file1` to destination `file2`. The destination file has a new creation date and inode number.
`cp` *`file* Dir1`*	Copies multiple files to a directory.
`cp -f` *`file1 file2`*	If you already have a file named `file2`, this command overwrites its contents without prompting.
`cp -i` *`file1 file2`*	If you already have a file named `file2`, this command prompts you for confirmation before overwriting this file.
`cp -p` *`file1 file2`*	Copies the contents of source `file1` to destination `file2`. The destination file has the same inode number and creation date as the source file.
`cp -r` *`Dir1 Dir2`*	Copies the contents of the directory named `Dir1`, including subdirectories, to `Dir2`. The effect is recursive; in other words, if there are subdirectories under `Dir1`'s subdirectories, their files and directories are also copied.
`cp -u` *`file1 file2`*	If you already have a file named `file2` and `file1` is newer, this command overwrites its contents without prompting.

NOTE *An* inode *is the identifier used on each Linux partition for a file. Every file gets its own inode. The inode includes metadata about the file, which includes the permissions, size, last access time, and the disk block where the file is located. If the inode is misaligned or corrupted, Linux won't be able to find the associated file. In addition, identical files have the same inode number. But because you can't have the same inode number on different partitions, the* `cp -p` *`file1 file2`* *command doesn't work if you're copying a file from one partition to another.*

mv

If you want to rename a file in Linux, you move it. The `mv` command changes the name of a file. Unless you're moving a file to a different volume, everything about the file, including the inode number, stays the same. There are four key move commands, as shown in Table 6.4.

Table 6.4: mv Commands

Command	Result
`mv` *`file1 file2`*	Changes the name of a file from `file1` to `file2`. If the source and destination files are located on the same volume, the files retain the same inode number.
`mv` *`file* Dir1`*	Moves multiple files to a directory.
`mv -f` *`file1 file2`*	If you already have a file named `file2`, this command overwrites its contents without prompting.
`mv -i` *`file1 file2`*	If you already have a file named `file2`, this command prompts you for confirmation before overwriting this file.

Tip *Some Linux users create files that start in lowercase, such as* `file1`*, and directories that start with a capital letter, such as* `Dir1`*. This is far from an absolute rule; standard Linux directories start in lowercase letters, such as* `/bin`*.*

rm

You can use `rm` to remove files and directories. This is one of the reasons many Linux administrators are advised to run Linux in root or superuser mode only when necessary; small mistakes in this command can easily delete all of your Linux files. For example, suppose you want to remove a group of temporary directories in your root (/) directory: `a.tmp`, `b.tmp`, and `c.tmp`. You want to use the `rm -r *.tmp` command, but instead you type the following:

```
# rm -r * .tmp
```

Because there's a space between the asterisk and `.tmp`, the shell assumes you want to recursively delete all directories and then delete the file named `.tmp`. The result is not good.

For this reason, Red Hat configures the following as an alias for the root user:

```
alias rm='rm -i'
```

The alias ensures that whenever you use the `rm` command (even `rm -r`), the shell prompts you for confirmation before you delete any files. Some Linux distributions set up this alias as a shell variable for root users. The key `rm` commands are shown in Table 6.5.

Tip *You can find default aliases with the* `alias` *command.*

ln

Instead of just copying or moving a file, you can link it. Links are common, especially for those programs that start at various runlevels. When you link a file, you're creating another path to a currently

TABLE 6.5: *RM* COMMANDS

COMMAND	RESULT
`rm` *`file1`*	Deletes *`file1`* without prompting for confirmation. However, this command does not supersede an `alias rm='rm -i'`, which requires confirmation.
`rm -d` *`Dir1`*	Deletes *`Dir1`* without prompting for confirmation. However, this command does not supersede an `alias rm='rm -i'`, which requires confirmation.
`rm -i` *`file1`*	Deletes *`file1`* after prompting for confirmation from the user.
`rm -f` *`file2`*	If you already have a file named *`file2`*, this command overwrites its contents without prompting. It even supersedes an `alias rm='rm -i'`.
`rm -r *`	Removes files recursively; if there are any subdirectories in the current directory, this command deletes them (and all of their files) as well. However, this command does not supersede an `alias rm='rm -i'`, which requires confirmation.

existing file. For example, if both you and a colleague are working on a file named `project`, you can create a linked file in your home directory. Assume the `project` file is in the `/home/jm` directory. To create a link to a file in `mj`'s home directory, you use the following command:

```
# ln /home/jm/project /home/mj/project
```

When you work on either file, the changes and results are visible and accessible to those who access both directories. This is sometimes known as a *hard link*. With a hard link, because both files retain the same inode number, both files are identical. If the original file is deleted, the hard-linked file remains in place. It retains all the information from the original file.

NOTE *The* `ln` *`file1 file2`* *command produces the same result as the* `cp -p` *`file1 file2`* *command. Unless the files are located on different partitions,* *`file1`* *and* *`file2`* *retain the same inode number.*

ADMINISTERING AS ROOT

One of the raging debates in the Linux community is whether it's sensible for a Linux administrator to log in as the root user. Errors as root can damage or destroy the files on your system. In addition, logging in as root may expose the root password to someone who has put a program on your system.

On the other hand, Red Hat has made it safer to use the root account. Good aliases make it more difficult to accidentally delete key files. Defaults such as `root_squash` in NFS prevent root users on other computers from sabotaging your system. You can further protect your system with passwords for the GRUB bootloader and your BIOS. Because the people I know at Red Hat use the root account regularly, I do the same in this book.

If you do log in as the root user, remember to be careful. Don't leave your system without logging out; otherwise, someone could change your password and access your system at his or her leisure. And don't expose your system to services that can read or even control what you do as root, such as the Virtual Network Computing (VNC) environment originally developed at AT&T (`www.realvnc.com`).

One useful option for links is *symbolic mode*, which allows you to see the linked file. For example, if you run the following command:

```
# ln -s /home/jm/project /home/mj/project
```

you will see the linked file when you run a long listing (`ls -l`) of that file. This is known as a *soft link*. If the original file is deleted, the soft-linked file points to an empty file. The information in the original file is lost.

mkdir and *rmdir*

As you'd expect, the `mkdir` command lets you create directories. The directory that you create does not have to be based in your current directory. You can make several levels of directories if you choose. You can also assign the permissions of your choice to the directory that you create. The key `mkdir` commands are shown in Table 6.6.

TABLE 6.6: *MKDIR* **COMMANDS**

COMMAND	RESULT
`mkdir -p Dir1/Dir2`	Creates a directory named *Dir2*. If *Dir1* does not exist, the `-p` switch tells Linux to create that directory as well. Both are created as subdirectories of the current directory.
`mkdir -m 755 /usr/Dir3`	Creates a directory named *Dir3* as a subdirectory in the `/usr` directory. The permissions (755) are `rwx` for the owner and `r-x` for other members of the group and everyone else.

The `rmdir` command allows you to delete empty directories. The directory you remove does not have to be based in your current directory. You can delete several levels of directories if the directory you delete empties others. For example, with the following command, you can delete the directories named *Dir1* and *Dir3*:

```
# rmdir -p Dir1/Dir3
```

This command deletes directory *Dir3* if it is empty. If the only "file" in directory *Dir1* is *Dir3*, this command also deletes directory *Dir1*.

Managing Files

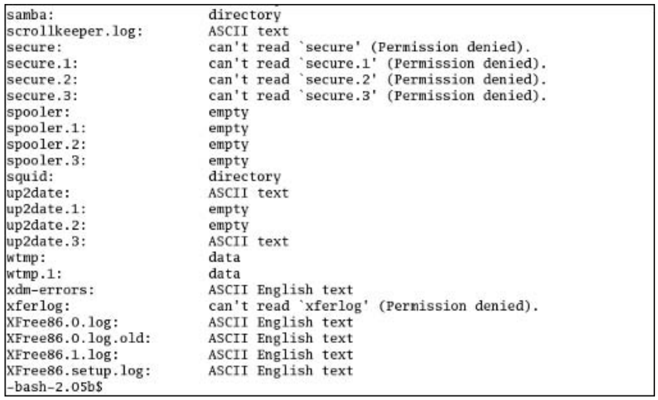
Linux includes a number of commands to help you read files in different ways. You can verify different types of files, and you can read files from the top or from the bottom. This read can be limited to a few lines, or it can set you up to page through the entire file. You can also count the lines, words, and alphanumeric characters within a file. In addition, Linux lets you search through a file using the search term of your choice.

Because it is difficult to define words or lines in binary files, most of these commands work best with text files.

file

Although some distributions differentiate between file types by color, there are no standard extensions in Linux. Files in Linux may or may not have extensions. Executable files don't end in `.exe`, and document files may not end in `.doc`. The `file` command allows you to view the type of each file. You can see how this works in Figure 6.2, where we ran the `file *` command as a regular user.

FIGURE 6.2 Reviewing different file types



```
samba:                  directory
scrollkeeper.log:       ASCII text
secure:                 can't read `secure' (Permission denied).
secure.1:               can't read `secure.1' (Permission denied).
secure.2:               can't read `secure.2' (Permission denied).
secure.3:               can't read `secure.3' (Permission denied).
spooler:                empty
spooler.1:              empty
spooler.2:              empty
spooler.3:              empty
squid:                  directory
up2date:                ASCII text
up2date.1:              empty
up2date.2:              empty
up2date.3:              ASCII text
wtmp:                   data
wtmp.1:                 data
xdm-errors:             ASCII English text
xferlog:                can't read `xferlog' (Permission denied).
XFree86.0.log:          ASCII English text
XFree86.0.log.old:      ASCII English text
XFree86.1.log:          ASCII English text
XFree86.setup.log:      ASCII English text
-bash-2.05b$
```

As you can see in Figure 6.2, you are not able to view the file type if you don't have the proper permissions.

cat

The concatenate (`cat`) command sends the text of a file to standard output. You can use the `cat` command on any file. The following command sends the text of the file to your screen:

```
# cat file
```

This command is flexible; you can even use it to read multiple files, in sequence, with the `cat` *`file1 file2`* command.

head and *tail*

The `head` and `tail` commands are like two sides of a coin. The `head` command provides you with a view of the first few lines of a file; the `tail` command provides you with a view of the last few lines of that same file. You can regulate the amount of the file you see with switches. For example, use the following command to see the first 15 lines of the `bully.txt` file:

```
# head -n15 bully.txt
```

If you substitute `tail` for `head`, you see the last 15 lines of this file. Table 6.7 lists more switches you can use with these commands.

TABLE 6.7: *HEAD* AND *TAIL* COMMANDS

COMMAND	RESULT
`head 400b bully.txt`	You see the first 400 bytes of the file known as `bully.txt`.
`tail 4k bully.txt`	You see the final 4KB of the file known as `bully.txt`.
`head 3m bully.txt`	You see the first 3MB of the file known as `bully.txt`.
`tail -n22`	You see the final 22 lines of the file known as `bully.txt`.

more and *less*

The `more` and the `less` commands aren't opposites, like `head` and `tail`. They both start at the beginning of a text file. When you run these commands on a text file, you review the contents of the file one page at a time. The `less` command is more versatile; unlike `more`, it allows you to scroll up and down any large text file by using the Page Up and Page Down keys on your keyboard.

Because they can read text a little bit at a time, these commands can open a file more quickly than a text editor such as `vi`. The `less` command also has some of the advantages of the `vi` editor, since you can use some `vi` commands to search through a file.

Each command includes two sets of options. A command such as the following sets up the file named `bigfile` with line numbers:

```
# less -N bigfile
```

Once the text file is open, you can run other commands, as described in Table 6.8.

TABLE 6.8: COMMANDS USED AFTER *LESS* IS APPLIED TO A TEXT FILE

COMMAND	RESULT
`space`	Pressing the spacebar on your keyboard scrolls forward one page in your screen.
`page up`	Scrolls back one page on your screen.
`page down`	Scrolls forward one page on your screen.
`#z`	# represents a number. For example, `8z` scrolls forward eight lines in the file. If you do not use a number, this command is equivalent to the `space` command.
`/abc`	Searches through the file for the text string `abc`. This is a command from the Linux `vi` text editor.

The `more` and `less` commands are also known as *pagers* because they allow you to review text files one page at a time using the Page Up and Page Down keys on your keyboard. When you've finished, just press the `q` key to exit from this "browse" mode.

Permissions

As shown in the output from `ls -l`, each file is associated with owners, groups, and a series of permissions. (For an example of this setup, see Figure 6.1.) The permissions associated with a file are

assigned to owners, groups, and everyone else on your Linux computer. Take a look at the following entry, which is the output from an `ls -l` command applied to a hypothetical file named `abc`:

```
-rwxrw-r-- 1 root root  1213 Feb 2 09:39 abc
```

Permissions are based on the characters on the far-left end of the output. The 10 characters determine what different users can do with this file.

If the first character is not a dash (`-`), it's not a regular file. It could be a directory (`d`) or a file that is linked (`l`) to another.

The remaining characters can be grouped in threes. The subsequent three characters shown are `rwx`. In other words, the owner of the file named `abc` can read (`r`), write (`w`), and execute (`x`) this file.

The next three characters shown are `rw-`. Users in the same group as the file owner can read this file (`r`) or edit and write to this file (`w`). These users can't execute the file.

The final three characters are `r--`. Users that don't belong to the same group as the file owner can read this file. They can't write to it, and they can't execute it if it's a script.

You can set up these permissions on any file using the following command:

```
# chmod 764 abc
```

Permissions are set with a three-number code. In the preceding command, the first number (`7`) sets permissions for the owner, the second (`6`) for the other users in the owners group, and the third (`4`) for everyone else. Each number represents all permissions given to the owner, group, or everyone else, as described in Table 6.9.

TABLE 6.9: NUMERIC PERMISSIONS

PERMISSION	NUMBER	BASIS
`r`	4	= r(4)
`w`	2	= w(2)
`x`	1	= x(1)
`rx`	5	= r(4) + x(1)
`rw`	6	= r(4) + w(2)
`wx`	3	= w(2) + x(1)
`rwx`	7	= r(4) + w(2) + x(1)

Look at the permissions associated with the file named `abc` again. Because the first number is `7`, the owner of this file has read (`r`), write (`w`), and execute (`x`) permission to this file. Since the second number is `6`, other users in the owner's group have read (`r`) and write (`w`) permissions on this file. Since the third number is `4`, everyone else has just read (`r`) permissions on this file.

TIP *Two closely related commands are* `chown` *and* `chgrp`*, which the root user can use to change the owner and group owner of a file. For example, the* `chown mj abc` *command makes the user mj the owner of the file* `abc`*.*

umask

When you create a new file or directory, the permissions you get depend on the value of what is known as the `umask`. Type `umask` at the command-line interface, and you'll see the current numeric *masked* value of your permissions.

```
# umask
0022
```

To understand this number, you need a clear idea of the numeric value of permissions. The first number in the `umask` is currently unused. So the actual `umask` is 022.

Now let's look at an example. If you gave everyone permissions to your files and directories, you would have read, write, and execute permissions for all users. As discussed in the previous section, these permissions correspond to the number 7 (`r+w+x` = 4+2+1). When applied to all users, they correspond to 777. You could set up the same permissions for all users on the `abc` file with the following command:

```
# chmod 777 abc
```

By convention, this corresponds to a `umask` of 000. However, `umask` does not allow you to configure execute (`x`=1) permissions on any file. Therefore, in reality, a `umask` of 022 corresponds to permissions of 644, or `rw-r--r--`; in other words, for new files, the owner has read and write permissions, the members of the group that own the file have read permissions, and all other users have read permissions.

Manipulating Files

Several commands are available that allow you to learn about and search for and through different files. The `wc` command allows you to get a count of the number of lines, words, and characters in a file. The `find` and `locate` commands let you search for specific files. The `grep` command enables you to search through a file for a text string without opening it. The `slocate` and `egrep` commands are variations on these commands.

wc

The `wc` command is fairly straightforward. With any text file, you have a certain number of lines, words, and characters. Using the `wc` command, you can find all three characteristics. For example, you can check the `showoff` text file as follows:

```
# wc showoff
1914   9298   76066
```

These numbers correspond to the number of lines, words, and characters in this file, respectively. You can get any individual figure based on the commands shown in Table 6.10.

TABLE 6.10: EXAMPLES OF THE *WC* COMMAND

COMMAND	RESULT
`wc -l showoff`	Number of lines in the file `showoff`
`wc -w showoff`	Number of words in the file `showoff`
`wc -c showoff`	Number of characters in the file `showoff`

find

The `find` command looks through directories and subdirectories for the file(s) of your choice. For example, if you want to find a file named `fig0606.tif`, you use the following command:

```
# find / -name fig0606.tif
```

This command searches in the root directory and all subdirectories for the `fig0606.tif` file. The search can take quite some time. If you have more information, you may want to substitute a lower-level directory for the root (`/`).

With the `find` command, you can also use wildcards, such as the asterisk (`*`) and question mark (`?`), in your search term.

locate and *slocate*

An alternative to `find` is the `locate` command. This command searches through an existing database of your files. By default, if you keep Linux running on your computer, the database associated with the `locate` command is refreshed every day at 4:02 A.M. If you're searching for a file that wasn't created since the last database update, the `locate` command finds files much more quickly.

In Red Hat Enterprise Linux, the `locate` command is actually soft-linked to the more secure `slocate` command. The database is updated per the `/etc/cron.daily/slocate.cron` script. Take a look at the second default command in that script:

```
/usr/bin/updatedb -f "nfs,smbfs,ncpfs,proc,devpts"
➥-e "/tmp,/var/tmp,/usr/tmp,/afs,/net"
```

As you can see from the `updatedb` man page, the `-f` switch excludes a number of filesystem types, and the `-e` switch excludes a number of directories that should be accessible only to the root user. You can customize this script to exclude other directories, such as `/root`, or filesystem types, such as `vfat`.

Once you have a `locate` database, it is more flexible; for example, if you use the following command, it returns all files that include the text string `fig0`:

```
# locate fig0
```

The `locate` command works as if asterisks are assumed before and after the search term.

grep

The grep command is a handy way to search through a file. As a system administrator, you may have long lists of users. If you want to search through your /etc/passwd file for a user named michael jang, try the following command:

```
# grep "michael jang" /etc/passwd
mj:x:500:500:michael jang:/home/mj:/bin/bash
```

This response tells you that there is a user named michael jang. It also includes the home directory and default shell for that user. If the search string exists in more than one line, you'll see those lines as well. You can even use grep to search through a series of files with commands such as the following:

```
# grep mj *
# grep -c bash /etc/passwd
```

The first command looks for the string mj in all files in the current directory. The second command, with the -c switch, counts the number of lines that include the word *bash*.

Command Combinations

It's a common practice to use more than one Linux command in a line. For example, if you're using the find command and you know that the result will have a large number of files, you can use a command such as grep to search through the result. Specifically, let's say you want to find some of the .html files on your system. You can start with the following command:

```
# find / -name *.html
```

However, you may get discouraged when you see hundreds of files flashing past you on your terminal screen. An alternative is to combine commands such as the following:

```
# find / -name *.html | grep bookmark
```

This command searches through the results of the find command for the text string bookmark. Only those files with both strings are output to the screen. Other possible command combinations include the following:

```
# who | grep mj
# ps aux | grep mozilla
```

The first command, who, lists all users currently logged onto your Linux system. When you pipe (|) the result to the grep mj command, you'll find the number of times that user mj is currently logged onto your system.

The second command, ps, lists the processes currently running on your Linux system. The three switches, aux (a dash is not required for ps command switches), leads to a very long list of processes, because it includes all processes run by all users (a), each associated with the username (u),

independent of the virtual terminal (`x`). You need a tool such as `grep` to search through these processes. This combined command returns all processes with the word *mozilla*, which is the default Red Hat graphical Web browser.

Using the *vi* Editor

Linux relies on a large number of text files for configuration. Therefore, you need a text editor to configure Linux. The `vi` editor may seem old. It certainly isn't the most popular editor, even in the Linux community. The one- or two-letter commands are cryptic, but if you ever need to rescue your system with a boot disk, `vi` may be the only editor at your disposal.

It is easy to open a file with `vi`. For example, if you want to open the `/etc/inittab` file, use this command:

```
# vi /etc/inittab
```

There are three basic ways to work in `vi`. Command mode is the default; you use insert mode when you want to insert text; and with a few special characters, execute mode can be used to run regular shell commands.

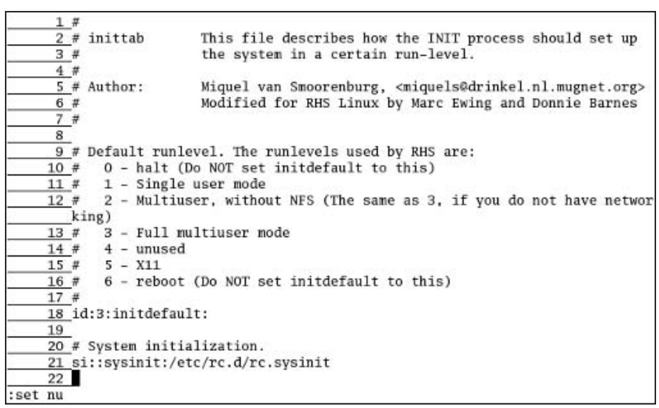
Command Mode

When you open a file in `vi`, the first mode is command mode. This is what you use to scroll through text, search for different text strings, or delete specific characters, words, or lines.

One aid in `vi` is line numbers, which you can activate by typing the following in the editor, which should lead to a result that looks similar to Figure 6.3:

```
:set nu
```

FIGURE 6.3
vi with line numbers



Getting Around

Although current versions of `vi` allow you to use the directional keys on your keyboard (arrows, Page Up, Page Down), this editor was designed for older U.S. keyboards that did not have these keys. Four lowercase letters take the place of the navigational arrows on the standard U.S. keyboard:

`h`	Left arrow
`j`	Down arrow
`k`	Up arrow
`l`	Right arrow

The alternatives to the Page Up and Page Down keys are Ctrl+B (back) and Ctrl+F (forward), respectively.

If you already know the line number you want, the `G` command can help. When used alone, it takes you to the last line in the file. When used with a line number, such as `20G`, it takes you to the desired line. As with Linux shells, case makes a difference, so make sure you're using the uppercase `G` for this command.

Deleting Text

It is easy to delete text in `vi`. Three deletion commands are associated with the current location of the cursor:

`x`	Deletes the current character, even if that character is a blank space or a tab
`dw`	Deletes the current word
`dd`	Deletes the current line

If you accidentally delete something, the `u` command reverses the last command entered.

Searching for Text

It is easy to search for text in `vi`. Just start with a forward slash. For example, if you want to search for the word *dollar* in a file, type the following:

```
/dollar
```

The cursor highlights the first place this word is found in the file. To proceed to the next instance of this word, type **`n`**. Just remember, case matters in a search in the `vi` editor.

Insert Mode

When you want to insert text into a file, use insert mode. There are several ways to do this, relative to the current location of the cursor (see Table 6.11).

In any case, getting out of insert mode is easy; just press the Esc key on your keyboard.

Table 6.11: Insert Mode Options

Command	Action	Comment
`i`	Insert	Everything you type is inserted, starting at the current position of the cursor.
`a`	Append	Everything you type is inserted, starting one character after the current position of the cursor. This is closely related to A (uppercase), where everything you type is inserted, starting at the end of the line with the cursor.
`o`	Open	Everything you type is inserted, starting one line below the current position of the cursor. Closely related is O (uppercase), where everything you type is inserted, starting one line above the current position of the cursor.
`cw`	Change word	Deletes the word (or space) that corresponds to the current position of the cursor. You get to insert text starting with that word.

Execute Mode

You can run regular shell commands from inside the `vi` editor. Just type `:!`, followed by the command. For example, if you were creating a script, you might need to know the directory location of a certain file. You could list the files in the `/etc/cron.daily` directory with the following command:

```
:!ls /etc/cron.daily
```

Regular execute mode starts with the colon (`:`). Several file management commands are associated with execute mode, including `:q` (to exit a file) and `:w` (to write the current text to the file). A number of basic commands for `vi` in all modes are shown in Table 6.12.

Tip *If you want to exit from* `vi` *without saving any changes, use the* `:q!` *command.*

Table 6.12: Basic *vi* Commands

Command	Description
`a`	Starts insert mode after the current cursor position.
`A`	Starts insert mode by appending at the end of the current line.
`cw`	Deletes the current word and then enters insert mode to allow you to replace that word.
`dw`	Deletes the current word without entering insert mode.
`dd`	Deletes the current line.
`G`	Moves the cursor to the end of the line.
`15G`	Moves the cursor to the 15th line.
`h`	Moves the cursor left one space.
`i`	Enters insert mode.

Continued on next page

TABLE 6.12: BASIC *VI* COMMANDS *(continued)*

Command	Description
`o`	Enters insert mode by opening a line directly below the current cursor.
`O`	The uppercase O command enters insert mode by opening a line directly above the current cursor.
`:q`	Exits from `vi`. If you have made changes and want to quit without saving, use `:q!`.
`r`	Replace; the next character you type replaces the current character.
`:set nu`	Activates line numbers for the current file.
`u`	Undoes the last change.
`:w`	Writes the current file.
Esc	Exits from insert mode.
`/system`	Searches for the word `system` in the current file.

Understanding Other Text Editors

Obviously, `vi` is not the only text editor available in Linux. Three other major text editors are `emacs`, `pico`, and `joe`. None of these editors is currently installed in Red Hat Enterprise Linux 3 server by default. The `pico` and `joe` editors aren't even included with Red Hat Enterprise Linux 3. However, you can still download and install them from the RPMs associated with Fedora Linux. Because this is not a book on text editors, we cover those three only briefly.

emacs

The `emacs` editor may be the most popular text editor used in the Linux/Unix world today. Once you've installed the `emacs` RPM, you can use it to open text files, just as you can with `vi`. For example, to open `/etc/inittab` in `emacs`, just run the following command:

```
# emacs /etc/inittab
```

NOTE *RPM is the Red Hat Package Manager, the standard way Red Hat organizes software; this system is covered in Chapter 10.*

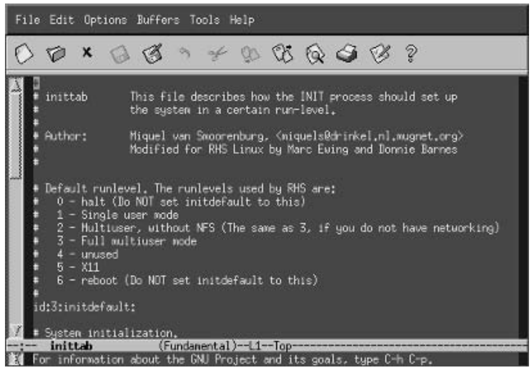
As you can see in Figure 6.4, opening `emacs` in a GUI brings up a menu-driven interface. If you want to know more about `emacs`, start the tutorial with the `Ctrl+h t` command.

pico

Another popular Linux/Unix editor is `pico`, which you can install separately as part of the `pine` e-mail RPM package. Once you've installed the `pine` RPM, you can use `pico` to open text files. For example, to open `/etc/inittab` in `pico`, just run the following command:

```
# pico /etc/inittab
```

FIGURE 6.4
The emacs editor

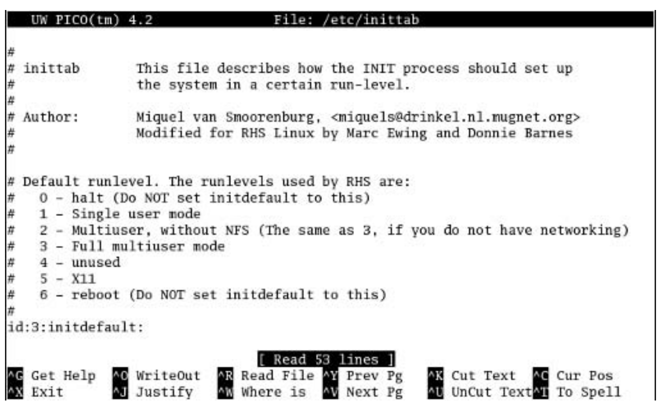


NOTE *The* `pine` *e-mail reader and* `pico` *editor is not included with Red Hat Enterprise Linux 3 or Fedora Linux. According to Red Hat, this is because the license associated with* `pine` *is not open source. If you want to use pine or pico, you'll have to use an RPM from another source such as Red Hat Linux 9.*

As you can see in Figure 6.5, opening `pico` in a GUI brings up a Ctrl character-driven interface. The Ctrl character, as shown in Figure 6.5, is a `^`. For example, the exit command shown is `^X`, which you can run with the Ctrl+`x` command.

Some of the available commands display at the bottom of the screen. As you can see, help and additional commands are available through the Get Help screen, which you can access with the Ctrl+`g` command.

FIGURE 6.5
The pico editor, no longer included with Red Hat Enterprise Linux 3



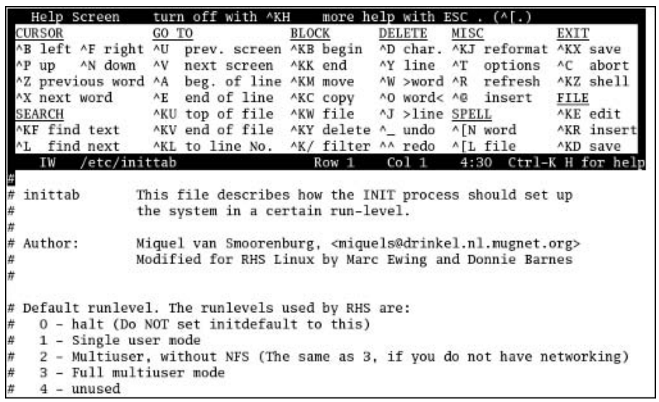
joe

Another popular Linux/Unix editor is `joe`, also known as "Joe's own editor." Once you've installed the `joe` RPM, you can use it to open text files. For example, to open the `/etc/inittab` file in `joe`, just run the following command:

```
# joe /etc/inittab
```

Opening `joe` in a GUI brings up a Ctrl character-driven interface. Unfortunately, the F1 key doesn't open help; the Ctrl+`k h` command is required. Some of the available commands display at the top of the screen, as you can see in Figure 6.6.

FIGURE 6.6 The joe editor, no longer included with Red Hat Enterprise Linux 3



Summary

In this chapter, we looked at some of the basic commands you can use at the command-line interface. Navigational commands can help you get around the Linux directory structure. Other commands can help you create, copy, move, delete, and link files and directories.

You can manage files by classifying their file types. You can also read the text in each file in a number of different ways. Linux lets you manipulate text files by counting lines, words, and characters; searching for specific files on your system; and searching for text within specific files. Command combinations help you focus on the information that you need.

Perhaps the most important Linux text editor is `vi`. While it is not the most popular text editor, it may be all you have available if you ever have to rescue your Linux system. The `vi` editor includes three modes: command, insert, and execute. Other major Linux editors include `emacs`, `pico`, and `joe`.

Now that you know some basic shell and `vi` editing commands, you're ready for Chapter 7, where you'll learn about the structure of Linux directories and the setup of some key configuration files. The next chapter also will help you learn how to manage, format, label, and troubleshoot hard disk partitions. With Logical Volume Management, you can even expand and contract virtual partitions to meet your needs.

Chapter 7

A Filesystem Primer

Everything in Linux is configured as a file. In Chapter 6, you worked with regular files and links to other files. As you learned, directories are just special types of files. In addition, hardware device drivers and partitions are represented by files. The organizational system for Linux files is known as the Filesystem Hierarchy Standard (FHS).

Filesystems are typically mounted on specific partitions. Linux servers often include several filesystems on different partitions. You can create partitions with `fdisk` or Disk Druid and format them to one of several standards. When you document the result in `/etc/fstab`, during the boot process Linux mounts the partitions as specified.

When you divide a hard drive into different partitions, you lose some flexibility; it isn't easy to expand the space available to a dedicated filesystem such as `/home`. The Logical Volume Management (LVM) system makes it possible to expand the size of a filesystem. This chapter covers the following topics:

- Understanding the Filesystem Hierarchy Standard
- Managing partitions
- Using formats and journals
- Mastering `/etc/fstab`
- Using the Automounter Alternative
- Exploring Logical Volume Management

Understanding the Filesystem Hierarchy Standard

When you install Linux, you can mount all Linux directories on a single partition. Alternatively, you can set up just about any Linux directory as a distinct filesystem by mounting it in a separate partition.

Establishing separate partitions limits risks to your system. For example, web servers such as Apache can accumulate log files that can consume gigabytes of space, easily crowding out all free space on your hard drive. Your users would no longer be able to save files, Linux would have no room to prepare print jobs, and chaos would undoubtedly result.

However, if you mount the right directory in a separate partition, your users can still work and save files even if the partition with the log files becomes full.

The Basic Linux Directory Structure

Before you select partitions for your Linux system, you first need to be familiar with the options in Linux directories. Red Hat Enterprise Linux organizes files into the following directories according to the Filesystem Hierarchy Standard (FHS):

`/` The top-level root directory. All other directories are below the root directory in the filesystem hierarchy. In other words, they are *subdirectories*. Any directory not mounted in a separate partition is automatically part of the root directory volume.

`/bin` Contains basic command-line utilities. You should not configure this directory in a separate partition. If you do, you won't be able to access these utilities in `linux rescue` mode.

`/boot` Includes the commands and files required for Linux to boot on your computer, such as the Grand Unified Bootloader (GRUB), the Initial RAM disk, and the Linux kernel. If you have a larger drive (more than 8GB), is generally a good idea to mount `/boot` in a separate partition. This helps ensure that your Linux boot files remain accessible when you start your computer.

`/dev` Lists available device drivers. For example, if you mount a floppy drive, you may mount `/dev/fd0` onto a directory such as `/mnt/floppy`. You should not mount this directory in a separate partition.

`/etc` Contains basic Linux configuration files, including those related to passwords, daemons such as Apache and Samba, and the X Window system.

`/home` Includes home directories for all but the root user. If you mount this directory in a separate partition, leave enough room for each of your users to add files.

`/initrd` Configures an empty directory used by the Initial RAM disk during the boot process. Do not mount this directory in a separate partition. If you delete this directory, Red Hat Enterprise Linux will not boot; you'll get a `kernel panic` message. This directory is not a formal part of the FHS.

`/lib` Lists program libraries needed by a number of different applications as well as the Linux kernel. You should not mount this directory in a separate partition.

`/lost+found` Contains orphan files. Utilities such as `fsck` place empty unidentifiable files (or parts of files) in this directory. This directory is not a formal part of the FHS.

`/misc` Notes a common mount point for shared NFS directories. This is also used by the Automounter, which we describe later in this chapter. This directory is not a formal part of the FHS.

`/mnt` Contains the mount point of removable media, such as floppy (`/mnt/floppy`), CD-ROM (`/mnt/cdrom`), and Zip (`/mnt/zip`) drives.

`/opt` Includes the standard locations for third-party applications such as Sun StarOffice or Corel WordPerfect.

`/proc` Includes all kernel-related processes currently running. Some of the files in this directory list current resource allocations; for example, `/proc/interrupts` lists currently allocated interrupt request (IRQ) ports.

`/root` The home directory for the root user. The `/root` directory is a subdirectory of the root (`/`) directory. Do not mount this directory separately.

`/sbin` Contains many system administration commands. Do not mount this directory separately.

`/tftpboot` Supports diskless workstations, also known as *remote terminals*. The diskless workstation mounts this directory from the Linux terminal server. This directory is not a formal part of the FHS.

`/tmp` Serves as a dedicated storage location for temporary files; this directory is also a good place to download files. By default, the `/etc/cron.daily/tmpwatch` script empties files older than 10 days from this directory.

`/usr` Includes programs and data available to all users; this contains many subdirectories. For example, the programs associated with the OpenOffice.org suite are installed in `/usr/bin`.

`/var` Contains variable data, including log files and print spools. On Linux servers, this directory is frequently mounted on a separate partition.

NOTE *The top-level root directory, `/`, is different from the home directory of the root user, `/root`. In fact, `/root` is a subdirectory of `/`.*

You'll want to mount some of these directories on separate hard drive partitions. For example, by mounting `/home` on a separate partition, you ensure that this directory will always have access to the space on that partition. In addition, by mounting `/var` on a separate partition, you can keep runaway log files from crowding out space needed by files in other directories. In the sections that follow, we discuss this approach in greater depth.

Partition Schemes

You now know that Linux provides a variety of ways to set up partitions. To help guide your efforts, there are a few standard partition schemes. By default, when you install Red Hat Enterprise Linux, you will set up at least two directories on separate partitions: the root (`/`) directory and `/boot`. The `/boot` directory is commonly mounted on its own partition because many Linux installations cannot start if the files in the `/boot` directory are stored above hard drive cylinder 1024.

NOTE *For some computers, you can configure the `/boot` directory above cylinder 1024 with LBA enabled; see Chapter 3 for more information. (LBA stands for Logical Block Addressing, which is the way a BIOS looks at the cylinders, heads, and sectors of a hard drive.)*

When you install Red Hat Enterprise Linux in the Server configuration, the default includes several more mounted directories: `/home`, `/usr`, and `/var`. Other configurations may be appropriate if

you're installing different Linux directories on different physical hard drives. Table 7.1 contains a short list of possible Linux partition configurations.

TABLE 7.1: POSSIBLE LINUX PARTITION CONFIGURATIONS

MOUNTED DIRECTORIES	COMMENT
`/`, `swap`	Typical configuration for a computer with one small hard drive.
`/`, `/boot`, `swap`	Typical configuration for a computer with a large hard drive. This is the default configuration for Red Hat Enterprise Linux 3.
`/`, `/boot`, `/var`, `swap`	Possible configuration where log file size, such as from a web server, is an issue. This can prevent runaway log files from crowding out all free space on your Linux computer.
`/`, `/boot`, `/home`, `swap`	Possible configuration for a Linux server with home directories for a number of other users. With other measures such as quotas, this can help regulate the amount of space taken by individual users.

Managing Partitions

When you partition a hard drive, you organize it into sections, which can then be formatted. Every hard drive requires at least one partition. In fact, you can divide a standard hard drive into 16 different partitions.

You can configure the following three types of partitions on an IDE or SCSI hard drive:

Primary partition You can create up to four different primary partitions on an IDE or a SCSI hard drive. One primary partition must be active; it should include a bootloader such as GRUB, the Linux Loader (LILO), or the Windows NT/2000/XP/2003 bootloader.

Extended partition If you need more partitions, you can convert one primary partition into an extended partition. The extended partition then can be further subdivided into logical partitions.

Logical partition An extended partition can be subdivided into the logical partitions that you need. You can have up to 12 logical partitions on a hard drive.

In Chapter 3, you used Disk Druid to create partitions during the Red Hat Enterprise Linux installation process. However, Disk Druid isn't available once Linux is installed; your only option is to use the `fdisk` utility.

Adding Partitions with *fdisk*

The `fdisk` utility is the traditional tool for managing partitions. While functionally similar to the MS-DOS tool of the same name, the Linux `fdisk` utility looks different and is much more versatile. It can help you manage the empty space on an existing drive. It lets you configure up to four primary partitions on a hard drive. You can use `fdisk` to change the type of partition to one of more than 100 types, including FAT32, Novell NetWare, Linux Logical Volume Manager, Linux Swap, and, of course, a standard Linux partition.

Adding a New Hard Drive

As a Linux administrator, you need to know how to add new hard drives to your servers. Once you've physically connected your hardware, make sure your PC recognizes it through your BIOS or other means. If your PC doesn't recognize it, there may be a problem with the new hard disk or the connections.

Once you've added a new hard drive, you need to set it up and configure it. The basic hard drive configuration utility is `fdisk`. Figure 7.1 illustrates a configuration with three physical hard drives.

Figure 7.1
fdisk shows three hard drives.

```
[root@Enterprise3 root]# fdisk -l

Disk /dev/hdd: 1073 MB, 1073741824 bytes
16 heads, 63 sectors/track, 2080 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdd1             1      2080   1048288+  83  Linux

Disk /dev/hda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1        13    104391   83  Linux
/dev/hda2            14       474   3702982+  83  Linux
/dev/hda3           475       522    385560   82  Linux swap

Disk /dev/hdb: 1073 MB, 1073741824 bytes
64 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot    Start       End    Blocks   Id  System
[root@Enterprise3 root]#
```

Note *While SCSI drives are more common, I've done most of my testing for this book using VMware. Unfortunately, the version of VMware workstation available as of this writing (4.5) doesn't handle SCSI on Linux very well.*

As you can see, the `fdisk -l` command lists partition tables on the local computer. This example has three IDE hard disks, designated `/dev/hda`, `/dev/hdb`, and `/dev/hdc`. The `/dev/hda` hard drive includes three partitions.

Note the number of cylinders in `/dev/hda`. Because that is the same number as the last cylinder of the last `/dev/hda` partition, you know that no room is available for additional partitions on the first IDE hard drive.

As shown in Figure 7.1, no partitions are associated with `/dev/hdb`. It's time to do something about that. Use `fdisk` to open the second IDE hard drive with the following command:

```
# fdisk /dev/hdb
```

Note *Depending on the value of your PATH variable, you may need to specify the full path to a command such as* `fdisk`*. If the* `fdisk` *command doesn't work by itself, try* `/sbin/fdisk`*. For more information on PATH, see Chapter 8.*

If this is a completely new hard drive, you'll see a message telling you the hard drive does not contain a valid partition table. If you don't see this message, it probably means someone has used the hard drive before. In either case, the next thing you'll see is the `fdisk` utility prompt:

```
Command (m for help):
```

Now press the `m` command to see the options available within the `fdisk` utility. The more important commands are described in Table 7.2.

TABLE 7.2: *FDISK* **COMMANDS**

COMMAND	RESULT
`a`	Sets or unsets the bootable flag. You need to make at least one primary partition on one of your first two hard drives bootable.
`d`	Deletes a partition. Before a partition is actually deleted, you need to select the partition number.
`l`	Lists known partition types. More than 100 different partition types are available.
`m`	Shows available `fdisk` commands.
`n`	Configures a new partition.
`p`	Lists the current partition table.
`q`	Exits `fdisk` without saving changes.
`t`	Allows you to change the partition system ID. You'll also need the partition number and the ID of the partition type you want, based on the known partition types (which you can find with the `l` command).
`v`	Verifies the current partition table.
`w`	Writes your changes and exits from `fdisk`. No changes are written to the partition table of your hard drive until you execute this command.

Now let's return to the task at hand: configuring a new hard drive. You're in the `fdisk` utility, and the first thing to do is to create a new partition. Issue the `n` command. The `fdisk` utility lets you choose whether you're creating a primary or an extended partition. If you already have an extended partition, `fdisk` allows you to create a logical partition.

```
Command (m for help): n
Command action
        e   extended
        p   primary partition (1-4)
```

Start by creating a primary partition with the `p` command. Make it the first primary partition, and start it with the first available cylinder. You can specify the size of the partition in cylinders, kilobytes, or megabytes. The sequence is shown in Figure 7.2. The first partition is configured as 100MB, starting with cylinder number one. As you can see, 100MB in this case corresponds to 49 cylinders.

You can continue this process until you've configured the space you need or you've allocated all the space on the new hard drive. Once you've finished configuring partitions, save your changes with the `w` command. If you want to start again, exit without saving by using the `q` command.

Before you can use your new partition, you need to format it to a system such as ext2, ext3, or VFAT. Details on this process are available later in this chapter.

Figure 7.2
Creating a new partition

```
[root@Enterprise3 root]# fdisk /dev/hdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklab
el
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-520, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-520, default 520): +100M

Command (m for help): p

Disk /dev/hdb: 1073 MB, 1073741824 bytes
64 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1             1        49     98752+  83  Linux

Command (m for help):
```

Managing an Existing Hard Drive

If you installed Red Hat Enterprise Linux on a large hard drive, you may have some extra space available. Remember, you can configure up to 16 partitions on your hard drive. In this section, we'll toggle a bootable partition, add a new extended partition, and then add a logical partition.

It's easy to make a partition bootable. Once in the `fdisk` utility, run the `a` command. Select the appropriate primary partition, and `fdisk` adds the bootable label. Figure 7.3 illustrates this process.

You can install a bootloader such as GRUB or LILO on a bootable partition on one of the first two hard drives on your computer.

Based on the configuration shown back in Figure 7.2, there are 471 free cylinders are still available on the new hard drive. The space under the Boot column is empty. And this is specifically labeled as a Linux partition.

Figure 7.3
Making a partition bootable

```
[root@Enterprise3 root]# fdisk /dev/hdb

Command (m for help): p

Disk /dev/hdb: 1073 MB, 1073741824 bytes
64 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1             1        49     98752+  83  Linux

Command (m for help): a
Partition number (1-4): 1

Command (m for help): p

Disk /dev/hdb: 1073 MB, 1073741824 bytes
64 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1   *         1        49     98752+  83  Linux

Command (m for help):
```

Now that you're more familiar with `fdisk`, creating extended and logical partitions is fairly easy. However, you can also select different cylinders. Figure 7.4 shows one set of commands you could use to create an extended and a logical partition.

FIGURE 7.4
Adding extended and logical partitions

```
[root@Enterprise3 root]# fdisk /dev/hdb

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
e
Partition number (1-4): 4
First cylinder (50-520, default 50): 200
Last cylinder or +size or +sizeM or +sizeK (200-520, default 520): 520

Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (200-520, default 200): 200
Last cylinder or +size or +sizeM or +sizeK (200-520, default 520): +300M

Command (m for help): p

Disk /dev/hdb: 1073 MB, 1073741824 bytes
64 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1   *         1        49     98752+  83  Linux
/dev/hdb4           200       520    647136    5  Extended
/dev/hdb5           200       345    294304+  83  Linux

Command (m for help):
```

Note how the cylinders of the first logical partition, `/dev/hdb5`, are contained within the cylinders of the extended partition. All logical partitions must fit within the space available to an extended partition.

Revising Partition Labels

You can use `fdisk` to configure partitions for swap space for LVM, or even for other operating systems.

When you use the Linux `fdisk` utility to create a new partition, it sets up the new partition with a Linux label by default. You can configure such partitions to the basic Linux formats: ext2, ext3, xfs, reiserfs, and so forth. However, there are a number of other ways to label a partition.

Use `fdisk` to open the hard drive with the partition you want to change. The following is based on Figure 7.4.

Now use the `t` command within `fdisk` to change the partition label. You'll need to select the partition number and then enter the hex code associated with the desired system. For example, the following commands changes the `/dev/hdb5` logical partition to the Linux swap system:

```
Command (m for help): t
Partition number (1-5): 5
Hex code (type L to list codes): 82
Changed system type of partition 5 to 82 (Linux swap)
```

As you can see in Figure 7.5, you can set a partition to be usable by a wide variety of operating systems.

FIGURE 7.5
Available *fdisk* partition systems

```
Command (m for help): l

 0  Empty           1c  Hidden Win95 FA 70  DiskSecure Mult bb  Boot Wizard hid
 1  FAT12           1e  Hidden Win95 FA 75  PC/IX           be  Solaris boot
 2  XENIX root      24  NEC DOS         80  Old Minix       c1  DRDOS/sec (FAT-
 3  XENIX usr       39  Plan 9          81  Minix / old Lin c4  DRDOS/sec (FAT-
 4  FAT16 <32M      3c  PartitionMagic  82  Linux swap      c6  DRDOS/sec (FAT-
 5  Extended        40  Venix 80286     83  Linux           c7  Syrinx
 6  FAT16           41  PPC PReP Boot   84  OS/2 hidden C:  da  Non-FS data
 7  HPFS/NTFS       42  SFS             85  Linux extended  db  CP/M / CTOS / .
 8  AIX             4d  QNX4.x          86  NTFS volume set de  Dell Utility
 9  AIX bootable    4e  QNX4.x 2nd part 87  NTFS volume set df  BootIt
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 b  Win95 FAT32     50  OnTrack DM      93  Amoeba          e3  DOS R/O
 c  Win95 FAT32 (LB 51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 e  Win95 FAT16 (LB 52  CP/M            9f  BSD/OS          eb  BeOS fs
 f  Win95 Ext'd (LB 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
10  OPUS            54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
12  Compaq diagnost 56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
14  Hidden FAT16 <3 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
16  Hidden FAT16    61  SpeedStor       a9  NetBSD          f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys ab  Darwin boot     fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden Win95 FA 65  Novell Netware  b8  BSDI swap       ff  BBT

Command (m for help):
```

Later in this chapter, you'll take the final steps to get new partitions ready for data. But first, partitions have journals that help Linux keep track of file locations on each partition.

Using Formats and Journals

As you saw in Figure 7.5, there are many ways to format a filesystem for different operating systems. In addition, there are several ways to format a partition for Linux. The latest versions of Linux include journaling features, which promote quick recovery from drive crashes. Each of these procedures set up different types of labels for your partitions.

Basic Linux Formats

As you've learned, you can format a filesystem in several ways. While the current default for Red Hat Enterprise Linux is the third extended filesystem, ext3, a number of other Linux filesystems are available that you may want to use. Table 7.3 lists the major Linux filesystem formats.

Linux is moving toward journaling filesystems such as ext3, reiserfs, and xfs. A journal records all pending changes, such as data to be written to disk. If a drive crashes, Linux can check the journal for pending changes. No disk check is required, which can save considerable time.

In the enterprise, the reiserfs and xfs filesystems are popular alternatives. They accommodate larger file sizes. For example, the current version of the xfs filesystem accommodates files as large as 9×10^{18} bytes, which is more than four million times the size of the largest allowable ext3 file. Red Hat has selected ext3 as its default in part to accommodate easy conversion from existing ext2 partitions. You can format a filesystem to a number of different filesystems after installation.

Several other Linux type filesystems are available, including ext, bfs, minix, and xia. None of these filesystems are commonly used on the Linux operating system today.

Table 7.3: Major Linux Filesystem Formats

Format	Description
ext2	The second extended filesystem, which was the standard for Red Hat Linux operating systems through 2001. If you have older systems with ext2 partitions, they're easy to convert to ext3.
ext3	The third extended filesystem, which is the current default for Red Hat Enterprise Linux 3. It includes a journal, which records all pending changes, such as data to be written to disk.
jfs	A journaling filesystem developed by IBM, which is still common on servers created by that company.
reiserfs	The Reiser filesystem, which is based on different designs from the Linux extended filesystems.
xfs	The filesystem developed by Silicon Graphics, which supports extremely large hard drives.

Formatting a Partition

Linux configures the `mkfs` command as a front end to format Linux partitions. If a partition has been previously formatted, all you need is that command, and Linux will reformat the partition to the same filesystem. Otherwise, you'll need to specify the type of filesystem to be built by including the `-t` switch. You can also check for bad blocks before formatting with the `-c` switch.

The commands are fairly straightforward. For example, the following commands format the first partition on a second SCSI hard drive, `/dev/sdb1`, to the noted filesystems:

```
# mkfs -t ext2 /dev/sdb1
# mkfs -t ext3 /dev/sdb1
# mkfs -t vfat /dev/sdb1
# mkfs -t reiserfs /dev/sdb1
```

Another way to create an ext3 filesystem is with the following command (the `-j` creates a journal):

```
# mkfs -j /dev/sdb1
```

Alternatively, if you're formatting a partition for Linux swap space, use the `mkswap` command. For example, if you want to set up `/dev/sdb5` as a swap partition, the following command is straightforward:

```
# mkswap /dev/sdb5
```

Tuning

It's easy to convert an older partition formatted to the ext2 filesystem to ext3. In fact, the ext3 filesystem is virtually identical to ext2. The only difference is that ext3 partitions include a journal.

Therefore, if you create a journal for an ext2 filesystem, it automatically becomes an ext3 filesystem. All you need is the `tune2fs -j` command. For example, the following command converts the `/dev/hda1` partition from ext2 to ext3:

```
# tune2fs -j /dev/hda1
```

Disk Management

Two similar disk management commands are available in Linux: `du` and `df`. The directory usage (`du`) command lists the amount of space used by each file in and below your current directory. The disk free (`df`) space command lists the amount of space available on each hard drive volume. Figure 7.6 shows the output you get if you run the `du` command in a Linux user's home directory.

FIGURE 7.6
Output from *du*

```
264     ./evolution/mail/pop/michael@ywow.org@mail.ywow.org
548     ./evolution/mail/pop
8       ./evolution/mail/pop3
560     ./evolution/mail
4       ./evolution/cache
16      ./evolution/meta/file
8       ./evolution/meta/vtrash
28      ./evolution/meta
28      ./evolution/config
8       ./evolution/private
86240   ./evolution
8       ./.elinks
12      ./.lftp
4       ./.emacs.d/auto-save-list
8       ./.emacs.d
4       ./Mail
7648    ./wks-cd
4       ./.gftp/cache
28      ./.gftp
72      ./.xvpics
264     ./comps/po
588     ./comps
102828  .
[root@Enterprise3 root]#
```

The number you see on the left is the size of the file, in kilobytes, and is the default from both the `df` and `du` commands. The applicable file displays on the right. For example, you may see the following:

```
86240   ./evolution
```

The first dot (`.`) means you start in your current directory. The slash (`/`) navigates to a subdirectory, in this case `evolution`. In other words, this line means there are 86,240 kilobytes of disk space dedicated to the `./evolution` subdirectory.

The `df` command shows how full each mounted filesystem is on your computer. As you can see in Figure 7.7, the `df -m` command assesses each filesystem and displays the results in megabytes. It includes any other filesystems, such as your floppy or CD-ROM drives that are currently mounted.

The `-m` switch gives you results in megabytes, and the `-k` switch gives you results in kilobytes.

FIGURE 7.7
Output from *df -m*

```
[root@Enterprise3 root]# df -m
Filesystem           1M-blocks      Used Available Use% Mounted on
/dev/hda2                 3560      3186       194  95% /
/dev/hda1                   99        15        79  16% /boot
none                       129         0       129   0% /dev/shm
/dev/hdd1                 1008       556       401  59% /home
[root@Enterprise3 root]#
```

Extended Partition Data

Linux includes a substantial amount of data with each partition, which you can access with commands such as `e2label` and `dumpe2fs`. When you install Red Hat Enterprise Linux, Linux partitions that you create during the installation process are automatically given appropriate label data. For example, I can get the following from this command:

```
# e2label /dev/sda1
/boot
```

As you will see later in this chapter, labels can be important. The default `/etc/fstab` uses disk labels. You can also find disk labels in the GRUB configuration file. But when you configure a new partition with `fdisk` and format it with `mkfs`, neither command adds a label. So if you want to mount the `/home/mj` directory on the `/dev/sdb1` partition, you should also label it with the following command:

```
# e2label /dev/sdb1 /home/mj
```

Alternatively, you can get more information about a partition with the `dumpe2fs` command. Look at Figure 7.8, which illustrates a partition with the `/boot` label.

FIGURE 7.8 Finding filesystem data with *dumpe2fs*

```
dumpe2fs 1.32 (09-Nov-2002)
Filesystem volume name:   /boot
Last mounted on:          <not available>
Filesystem UUID:          ae6d740d-25fb-40d2-b2e7-a5585f8345af
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal filetype needs_recovery sparse_super
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              26104
Block count:              104391
Reserved block count:     5219
Free blocks:              86054
Free inodes:              26058
First block:              1
Block size:               1024
Fragment size:            1024
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         2008
Inode blocks per group:   251
Filesystem created:       Thu Oct 23 11:17:24 2003
Last mount time:          Tue Mar  2 15:40:01 2004
Last write time:          Tue Mar  2 15:40:01 2004
```

You can even check the last time this partition was mounted with the Last Mount Time variable from the `dumpe2fs` output; if you see a `n/a` on that line, no directory has been mounted on this platform.

Mounting Directories

Before you can read or write to a Linux partition, you need to mount it. Without any help, you need to specify the partition, the directory being mounted, and the format associated with the partition. A typical syntax of the `mount` command is as follows:

```
# mount -t format partition directory
```

The *`format`* is the way the partition is configured, such as ext3, reiserfs, or xfs. The *`partition`* is the hard drive device being mounted, such as `/dev/sda1` or `/dev/hda1`. And the *`directory`*, also known as the *mount point*, is the part of the Linux directory structure allocated to that partition, such as `/boot`, `/home`, or `/var`.

In other words, you could mount the `/home/mj` directory on the `/dev/sdb1` partition that has been formatted to the ext3 filesystem with the following command:

```
# mount -t ext3 /dev/sdb1 /home/mj
```

This is more complicated than is normally required. With the list of formats in the `/etc/filesystems` configuration file, the `mount` command can look through this file and find a format that matches the `/dev/sdb1` partition. So all you need is the following command:

```
# mount /dev/sdb1 /home/mj
```

You can make this even simpler. If you add the following line to your `/etc/fstab` configuration file, you need to specify only the partition or the directory:

```
/dev/sdb1   /home/mj      ext3  defaults  1 2
```

Once configured in `/etc/fstab`, either of the following commands would work:

```
# mount /dev/sdb1
# mount /home/mj
```

Sometimes it's important to unmount a directory. For example, Linux locks the CD drive on many computers until you unmount the relevant directory with a command such as the following:

```
# umount /mnt/cdrom
```

Note that this command is spelled `umount`, not *unmount*.

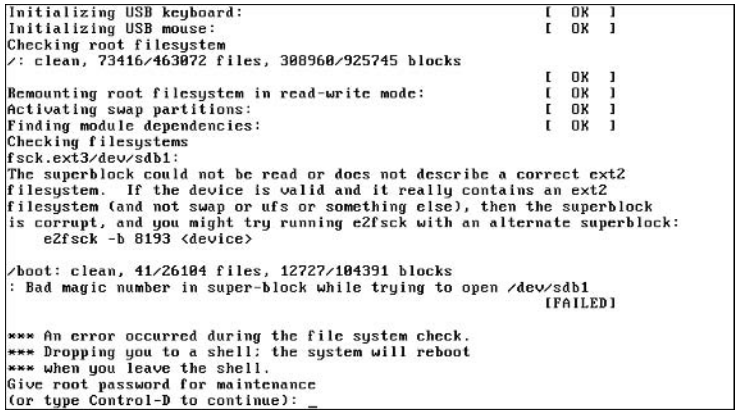
Troubleshooting

The failure of a filesystem can be more troubling than problems booting Linux. As you'll see in Chapter 11, there are established methods for getting around boot problems. However, filesystem problems are more difficult to diagnose. They can be a sign of corrupted files, misaligned blocks, troubled configuration files, or even bad hardware.

Filesystem problems usually require troubleshooting during the boot process. Linux may have trouble when mounting a specific partition or when a check of the filesystem integrity fails in some way. In either case, you may see a message that the `fsck` operation failed and that you need to type in the root password to gain access to Linux. An example of this situation is shown in Figure 7.9.

The `fsck` command is an important tool. Linux uses it periodically to automatically check most of the partitions on your system. If you don't have a filesystem integrity problem, you may just need to adjust a parameter and remount a filesystem, such as the root (`/`) directory.

FIGURE 7.9
Linux's response to a filesystem problem



```
Initializing USB keyboard:                                 [  OK  ]
Initializing USB mouse:                                    [  OK  ]
Checking root filesystem
/: clean, 73416/463072 files, 308960/925745 blocks
                                                           [  OK  ]
Remounting root filesystem in read-write mode:             [  OK  ]
Activating swap partitions:                                [  OK  ]
Finding module dependencies:                               [  OK  ]
Checking filesystems
fsck.ext3/dev/sdb1:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

/boot: clean, 41/26104 files, 12727/104391 blocks
: Bad magic number in super-block while trying to open /dev/sdb1
                                                           [FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell: the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue): _
```

FSCK

The `fsck` command checks and repairs Linux filesystems. As with `mkfs`, it is a front end to commands that are dedicated to relevant filesystems, such as `fsck.ext2`, `fsck.ext3`, and `fsck.reiserfs`. If the filesystem format is known, the `fsck` command is all you need. If the partition is formatted to `ext3`, the `fsck.ext3` command is called automatically. Table 7.4 describes several key options for this command.

TABLE 7.4: *FSCK* **COMMAND OPTIONS**

SWITCH	EXPLANATION
`-a`	Automatically repairs target filesystems without prompts. Should be used only within `/etc/rc.sysinit`.
`-b` *`superblock`*	Uses a different superblock. You can find optional superblocks via the `dumpe2fs` command.
`-A`	Checks all filesystems listed in `/etc/fstab`.
`-R`	When `-A` is used, skips the root (`/`) directory filesystem.
`-y`	When `fsck` suggests a solution, it sets a default answer of "yes."

WARNING *Don't run* `fsck` *on a mounted partition. It can lead to severe filesystem damage.*

If you suspect a problem, you can run `fsck` on any *unmounted* partition. Generally, you should accept the default suggestions for repairing any filesystem problems. While some data may be lost, this process should make your partition bootable again. At that point, you should be able to reboot Linux cleanly.

NOTE *Incidentally, the pronunciation of* `fsck` *varies; some may say* ef es check, *while others may talk about running the* fisk *command on a partition.*

Automated Partition Checks

The `fsck` command is no longer run periodically by default. However, you can change this by using the `tune2fs -c` *`count`* `/dev/`*`partition`* command. To find the mount count information for a specific partition, use the `dumpe2fs` command. The relevant output is shown in Figure 7.10.

Figure 7.10
Periodic partition check information

```
Last mount time:          Tue Mar  2 15:40:01 2004
Last write time:          Tue Mar  2 15:40:01 2004
Mount count:              70
Maximum mount count:      24
Last checked:             Sat Dec 13 20:56:39 2003
Check interval:           15552000 (6 months)
Next check after:         Thu Jun 10 21:56:39 2004
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal UUID:             <none>
Journal inode:            8
Journal device:           0x0000
First orphan inode:       0


Group 0: (Blocks 0-32767)
  Primary superblock at 0, Group descriptors at 1-1
  Block bitmap at 2 (+2), Inode bitmap at 3 (+3)
  Inode table at 4-515 (+4)
  10587 free blocks, 16370 free inodes, 2 directories
  Free blocks: 22181-32767
  Free inodes: 15-16384
Group 1: (Blocks 32768-65535)
--More--
```

Mounting and Remounting

One of the options in the rescue modes discussed in Chapter 11 is to mount filesystems such as root (/) in read-only mode. Once you've made the necessary changes, you can mount that filesystem in read-write mode. Alternatively, you may want to mount a filesystem with programs such as `/usr` in read-only mode.

Eventually, you may want to change the `/etc/fstab` configuration file and reboot Linux. However, you can test your changes first by remounting the directory with the desired options. For example, if you want to remount the root (/) directory in read-write mode, use the following command:

```
# mount -o remount,rw /
```

Or if you want to remount `/usr` in read-only mode, you could use the following command:

```
# mount -o remount,ro /usr
```

Remember, changes with this command apply only until you reboot Linux, unless you also revise the `/etc/fstab` file accordingly.

Mastering */etc/fstab*

Linux uses `/etc/fstab` during the boot process to mount partitions and directories in different ways. As shown in Figure 7.11, a number of parameters are associated with each filesystem. These parameters determine how filesystems are mounted, the way data is read, which user permissions are associated with the filesystem, and more.

FIGURE 7.11
/etc/fstab defines how directories are mounted.

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /proc                   proc    defaults        0 0
none                    /dev/shm                tmpfs   defaults        0 0
/dev/hda3               swap                    swap    defaults        0 0
/dev/hdd1       /home           ext3    defaults,usrquota,grpquota      0 0
/dev/cdrom              /mnt/cdrom              udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0                /mnt/floppy             auto    noauto,owner,kudzu 0 0
~
~
"/etc/fstab" 9L, 674C
```

As you can see, each `/etc/fstab` line includes six fields. Table 7.5 describes these fields, which are listed from left to right.

TABLE 7.5: */ETC/FSTAB* **FIELDS**

COLUMN	FIELD	DESCRIPTION
1	Label	The filesystem, such as `/usr`, or partition, such as `/dev/sdb1`, to be mounted.
2	Mount Point	The directory where the partition or filesystem is to be mounted.
3	Format	The filesystem format type, such as `ext2`, `ext3`, or `reiserfs`.
4	Mount Options	The defaults option includes `rw` (read-write), `suid` (SUID permissions), `dev` (terminals and block devices such as drives), `exec` (binary files), `auto` (automatically mounted), `nouser` (only root can mount), and `async` (data is read and written asynchronously).
5	Dump Value	If 1, the filesystem is automatically written to disk.
6	Filesystem Check Order	Filesystems that need `fsck`. The root (/) filesystem should be 1; others on the local computer should be 2; swap, virtual, CD, floppy, and remote directories should be 0.

In most cases, a listing such as `LABEL=/` is checked against the partition data on your computer to find the actual partition device, such as `/dev/hda3`, to be mounted.

Other mount options are available, such as `usrquota` and `grpquota` (for setting quotas), `noauto` (to make sure Linux doesn't look for a CD or floppy when it boots), and `user` (to let any user mount a filesystem, such as a CD-ROM).

Using the Automounter Alternative

With regularly mounted partitions, the filesystem stays mounted until you run the appropriate `umount` command. This can be problematic. If you remove media such as a tape drive, Linux may not have finished writing files to that drive. If you have problems with a network connection, Linux may not have written the files to the network server before the connection was broken. Even worse, a broken network connection can "hang" a workstation, especially with an NFS mounted directory.

With the Automounter, you can configure temporary access to removable and network filesystems on an "as-needed" basis. The automounted directory is unmounted automatically after a fixed period of inactivity. Naturally, any files are written to that directory before unmounting.

Basic Configuration Files

There are two key configuration files associated with the Automounter on Red Hat Enterprise Linux: `auto.master` and `auto.misc`, both in the `/etc` directory. By default, these configuration files support the mounting of automounted directories on the `/misc` directory.

Naturally, the governing configuration file is `auto.master`, which includes the following command, which you can activate by removing the pound sign (`#`):

```
# /misc  /etc/auto.misc   --timeout=60
```

This points to the `auto.misc` file for the actual configuration; any automounted directories are unmounted after 60 seconds of inactivity. Naturally, you can use this same command to set up the Automounter on other directories. For example, if you don't already have a `/home` directory configured on your computer, you could configure an automatic mount to a `/home` directory on a remote server. With the following command in `/etc/auto.master`, the `/home` directory is automatically unmounted 30 seconds after the user on that workstation logs off the network:

```
/home  /etc/auto.setup --timeout=30
```

In Red Hat Enterprise Linux 3, the Automouter is run by the `autofs` daemon by default. You can start it and set it up to run in standard runlevels with the following commands:

```
# service autofs start
# chkconfig --level autofs 35 on
```

For more information on the `service` and `chkconfig` commands, see Chapter 13. (The pound sign is normally used as a command prompt in this book; it is also used to comment out a command in a configuration file.) For more information on the `service` and `chkconfig` commands, see Chapter 13.

Sample Setup

The key Automounter configuration file is `auto.misc`. In several cases, you'll need to modify the commands before activating them. For example, this file refers to ext2; the default filesystem format for Red Hat Enterprise Linux is ext3. The default version of this file is already preconfigured with the following command, which can work with your CD drive:

```
cd  -fstype=iso9660,ro,nosuid,nodev  :/dev/cdrom
```

This supports access using the standard CD filesystem, ISO9660. It configures the filesystem as read-only (`ro`), special user ID permissions are not allowed (`nosuid`), and devices are not allowed (`nodev`) either. If you've configured `auto.master` and activated the `autofs` service, this allows you to read the files on your CD with the following command:

```
#ls /misc/cd
```

You can configure the Automounter for a number of other systems. The default `auto.misc` file provides a number of examples, which are commented out with a (#) by default. If you activate the following command:

```
#linux   -ro,soft,intr   ftp.example.org:/pub/linux
```

you'll be able to access a shared NFS directory, `/pub/linux`, from the ftp.example.org computer, as read-only, with the `ls /misc/linux` command. If you don't commonly access the `/boot` directory, and it's configured on `/dev/hda1`, you can activate the following command:

```
#boot -fstype=ext2  :/dev/hda1
```

You can then access the `/boot` directory with the `ls /misc/boot` command. Naturally, you should change ext2 to ext3, assuming you use the default Red Hat filesystem format. In the same manner as with the CD, you can set up an automatic mount of a floppy drive by activating one of the following three commands:

```
#floppy    -fstype=auto  :/dev/fd0
#floppy    -fstype=ext2  :/dev/fd0
#e2floppy  -fstype=ext2  :/dev/fd0
```

The first command is the most flexible, as it tries to match the format of your floppy with `/etc/filesystems` to find the most appropriate format. You should make sure the other commands match the actual filesystem format of your floppy, probably ext3 or vfat. The following command is designed for a removable drive, connected as the third SCSI drive on your system:

```
#jaz  -fstype=ext2  :/dev/sdc1
```

The last sample command is configured for a removable IDE hard drive attached to the slave port of the secondary IDE controller. If it's connected and active, you'll be able to access this drive with the `ls /misc/removable` command

```
#removable  -fstype=ext2  :/dev/hdd
```

Exploring Logical Volume Management

Without Logical Volume Management (LVM), the decision of how to partition your hard drives during the Red Hat Enterprise Linux installation is critical. Once drives are partitioned, there are no easy way to expand the available space.

For example, assume you've set up the `/home` directory in a separate partition. You've planned ahead and assumed that you'll have enough space for 10 users. But your company expands, and suddenly, you need space on `/home` for 20 users. Without LVM, there is no easy way to expand the size of the `/home` partition. You'd need to back up the files from `/home`, find a partition with the space you need, and then restore the files to that new partition.

LVM allows you to reallocate chunks of disk space between different filesystems. So with LVM, if you have extra room in a filesystem such as `/var`, you can reallocate that space to `/home`.

You can create an LVM volume group during the installation of Red Hat Enterprise Linux, as described in Chapter 3. You can also create and manage an LVM volume group using the techniques described in the following sections. Even if you've already created an LVM volume group when you installed Red Hat Enterprise Linux, read on. LVM doesn't help unless you can use the commands described in the following sections to increase and decrease the size of your *logical volumes*.

Fundamentals

LVM is essentially a mapping of different physical sections of a hard drive. Once collected into a logical volume, filesystems such as `/home` and `/usr` can be mounted on that volume. You can reorganize logical volumes to include additional hard drive space.

That's the short version of what you can do with LVM. To really understand what happens in LVM, start with the following fundamental definitions:

Physical volume (PV) A *physical volume* (PV) usually corresponds to a standard primary or logical partition on a hard drive.

Physical extent (PE) A *physical extent* (PE) is a chunk of disk space. Physical volumes are divided up into a number of equal sized PEs.

Logical extent (LE) A *logical extent* (LE) is a chunk of disk space. The size of an LE in an LVM system is the same as the size of PEs on that system. Every LE corresponds to a specific PE.

Logical volume (LV) A *logical volume* (LV) is a collection of LEs. You can mount filesystems such as `/usr` and `/boot` on an LV.

Volume group (VG) The LVs on your system, collected together, form a *volume group* (VG). When you configure an LVM system, most commands are applied to a VG.

Creating a Physical Volume

If you're implementing LVM for the first time, it may be more convenient to configure it on a new hard drive. After installing the drive, don't install any partitions on it yet. You can create a PV on an entire hard drive. For example, if the hard drive is the slave on the secondary IDE connector, the Linux device is `/dev/hdd`. To create a PV on that disk, run the following command:

```
# pvcreate /dev/hdd
```

Alternatively, if you've already set up partitions with a utility such as `fdisk`, you can set up PVs on specific partitions. First, run `fdisk` to change the system ID of the desired partition. Once you're in the `fdisk` menu, the following commands would change hypothetical partition number 10 on a hard drive:

```
Command (m for help): t
Partition number (1-15): 1
Hex code (type L to list codes): 8e
```

Don't use this command on any partition where you want to keep the data. Once the type is changed to Linux LVM, you can then create a physical volume with a command such as the following:

```
# pvcreate /dev/hdd1
```

Once you've configured two or more PVs, the next step is to create a volume group.

Creating a Volume Group

A VG is a collection of PVs that are configured on one or more hard drives. You can create a VG from existing PVs. When you add more PVs, you can add them to existing VGs.

It's easy to create a VG. You can even give that VG the name of your choice, such as `programs`, with a command such as the following:

```
# vgcreate programs /dev/sdc1 /dev/sdd1
```

Once you have a VG, it's easy to add PVs with the following slightly different command:

```
# vgextend programs /dev/sde1
```

Now you can organize a VG into chunks that you can set up in a PV.

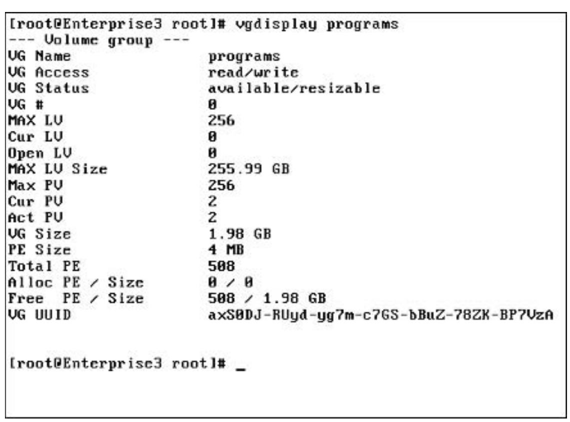
Creating a Logical Volume

Finally, you can create a logical volume where you can mount a filesystem such as `/home` or `/var`. But first, you need to know the size of a PE in your volume. You can do this and more with the `vgdisplay` command. Using the VG created in the previous section, this requires a command such as the following:

```
# vgdisplay programs
```

A sample output is shown in Figure 7.12. As you can see, this includes information on the maximum number of logical and physical volumes for this group, the size of this volume group, and the size of PEs in this group (in the figure, it's 4MB).

FIGURE 7.12
Volume group details



```
[root@Enterprise3 root]# vgdisplay programs
--- Volume group ---
VG Name               programs
VG Access             read/write
VG Status             available/resizable
VG #                  0
MAX LV                256
Cur LV                0
Open LV               0
MAX LV Size           255.99 GB
Max PV                256
Cur PV                2
Act PV                2
VG Size               1.98 GB
PE Size               4 MB
Total PE              508
Alloc PE / Size       0 / 0
Free  PE / Size       508 / 1.98 GB
VG UUID               axS0DJ-RUyd-yg7m-c7GS-bBuZ-78ZK-BP7VzA

[root@Enterprise3 root]# _
```

Now you can create an LV of the size that you need, with a command such as the following:

```
# lvcreate -l num_of_PEs programs -n logicvol
```

From the previous example, the name of the new LV is `logicvol`. You know that each PE is 4MB in size. If you wanted to set up a 200MB `logicvol` LVM partition, substitute `50` for ***`num_of_PEs`***.

This creates a new device, `/dev/programs/logicvol`. You can now format and mount that device just like any other hard drive partition. For example, the following commands format it to the ext3 filesystem and mount it on the `/tmp` directory:

```
# mkfs -j /dev/programs/logicvol
# mount -t ext3 /dev/programs/logicvol /tmp
```

Now it's easy to increase the size of `/dev/programs/logicvol`. Assuming you have spare PEs, just use the `lvextend` command. The following example increases the size of `/dev/programs/logicvol` to 300MB:

```
# lvextend -L300M /dev/programs/logicvol
```

Summary

In this chapter, we examined how files and filesystems work in Linux. Files are organized in a distinct structure known as the Filesystem Hierarchy Standard (FHS). Different directories in the FHS have different functions; you can mount many of these directories on their own partitions.

The basic Linux partition management utility is `fdisk`. With this utility, you can manage the empty space on existing or newly installed hard drives. You can create and size new partitions, and you can set or change them for different Linux formats.

Once you have a new partition, you can format it with the `mkfs` command. It's easy to format a partition to the Red Hat Enterprise Linux standard format, the ext3 filesystem. Just use the `mkfs -j` command. You can even convert an existing ext2 partition to an ext3 partition by using the `tunefs -j` command. Red Hat Enterprise Linux uses `fsck` to troubleshoot partitions on a regular basis.

The key filesystem configuration file is `/etc/fstab`, which defines how different partitions are mounted and checked.

If you need to configure temporary mounts, the Automounter can help. With proper configuration in `/etc/auto.misc`, you can set up temporary mounts on the removable and network directories of your choice.

Now with Logical Volume Management, you can vary the size of a partition based on the way you configure partitions into volume groups.

Now that you understand the basics of filesystems, the next chapter continues your exploration of the shell. You'll learn all the details you need to make the shell work effectively for you.

Chapter 8

Making the Shell Work for You

In the previous two chapters, we examined many of the fundamental commands that you need to navigate and administer Red HatEnterprise Linux. In this chapter, you'll learn the tricks of the trade that can help make the shell work for you.

The default Red Hat Enterprise Linux shell is bash (short for the Bourne Again Shell). While several other shells are available, bash is the default shell created by the Free Software Foundation (`www.fsf.org`) and is therefore the shell most commonly associated with Linux.

If you are already familiar with a different shell such as Korn, C, or Z, install the applicable RPM packages and use that shell. It's best to configure Linux in the language with which you're most familiar. They're easy to start; the `ksh`, `csh`, and `zsh` commands start these shells automatically, usually at a different prompt. Like other Unix-style operating systems, Linux works well with other shells. However, if Linux is your first foray into Unix-style operating systems, I highly recommend that you learn to use bash. It is the default Linux shell, and most online Linux documentation assumes that you use bash commands.

In this chapter, you'll learn to manage the basics of bash. Then you'll examine the secrets of the shell, which can help you make different bash commands work together in a complex harmony. You'll take advantage of *environment* and *shell* variables, which can ease your transition to the bash shell. Finally, you'll explore the world of scripts, which can ease your work as a Linux administrator. This chapter covers the following topics:

- Managing the shell
- Configuring the shell
- Discovering the secrets of the shell
- Creating basic scripts

Managing the Shell

The Bourne Again Shell (bash) is a user interface to the Linux operating system. You use bash commands to run programs, manage your files, and interact with your hardware through the Linux kernel. You can configure bash with a number of local and systemwide files and variables.

Shells such as bash are also known as *command-line interpreters*, which is a user interface that responds to specific commands, such as `ls`, `cd`, or `cp`. Shells also respond to programs or scripts that you create.

As you move around the command line, keep in mind that Linux is case sensitive. In other words, the `ls` command lists the files in your current directory, whereas the `LS`, `Ls`, or `lS` commands are meaningless in any current Linux shell.

Two ways the bash shell can help you are based on its history of previous commands and the ease with which you can complete a longer command. These characteristics are known as *interactivity* and *command completion*.

Interactivity

Interactivity allows you to run through previous commands. It also allows you to interact with current commands. You can use basic keys, such as the Home key and the four arrow keys, to correct typos; alternatively, you can even use commands you've used in a text editor.

Interactivity and History

You can interact with a history of Linux commands. Open a command-line interface and type the `history` command. If you've previously used the command-line interface, you'll see a result similar to that shown in Figure 8.1.

Figure 8.1
A history of previous commands

```
  511  emacs /etc/inittab
  512  poweroff
  513  history
  514  vi /boot/grub/grub.conf
  515  fdisk -l
  516  rpm -q kernel-source
  517  vi /etc/fstab
  518  chkconfig --list nfs
  519  chkconfig --list crond
  520  redhat-config-xfree86
  521  lpr /etc/passwd
  522  ./netsrc
  523  grub &
  524  gimp &
  525  service network status
  526  ls /etc/rc.d/init.d/
  527  cat /etc/inittab
  528  history
  529  service ypbind status
  530  chkconfig --list portmap
  531  chkconfig --list tux
  532  service tux status
  533  history
[root@Enterprise3 root]#
```

By default, you can repeat previous commands in several ways. The easiest way is to use the Up and Down arrow keys on your keyboard. Go to your command-line interface. When you press the Up arrow key, you'll see the previous commands you used, in reverse order. The list may even include commands you used during previous sessions. You can reverse the process with the Down arrow key.

Alternatively, if you remember the first letter of a recent command, use an exclamation point (`!`) to recall that earlier command. For example, based on the output of the `history` command shown in Figure 8.1, if you type `!r` the shell recalls the last time you used a command that started with the letter *r*—in this case, `redhat-config-xfree86`—and runs that command.

This feature is flexible; you can add a bit more information, such as `!rp`. Based on the history shown in Figure 8.1, the shell would respond by running the `rpm -q kernel-source` command. You

can even cite the number in your command history; for example, if you type `!512`, the shell recalls the `poweroff` command.

The feature lets you go back quite a bit. If you type the `env | more` command, you should find a `HISTSIZE=1000` line, which means you can go back and rerun any of the past 1,000 commands.

Interactivity and Editors

You can also interact with the details of a current or a previous command. For example, take the following command, which includes a typographical error:

```
# rpm -Vvh /mnt/cdrom/RedHat/RPMS/sendmail-*
```

You realize that you should have typed the `rpm -Uvh` command, but you don't want to retype the entire command. Fortunately, you don't have to erase the entire command. You can use basic keys such as the Left and Right arrows and the Home key to move the cursor toward the beginning of the command.

Alternatively, you can use the commands that you know in a text editor. For example, if you want to set `vi` as the default command-line editor, run the following command:

```
# set -o vi
```

NOTE *The* `set` *command is counterintuitive; while* `set -o` ***`editor`*** *enables that* ***`editor`***, `set +o` ***`editor`*** *disables it. For this command,* ***`editor`*** *can be* `emacs` *or* `vi`.

Now you can use `vi` editor commands. By default, you're in insert mode at the command-line interface. As we discussed in Chapter 6, you can switch to command mode by pressing the Esc key. Then you can apply the `vi` commands of your choice to that line. Some useful `vi` commands not described in Chapter 6 are shown in Table 8.1.

Table 8.1: More *vi* Commands

Command	Description
Home	Moves to the beginning of a line
b	Moves left one word
w	Moves right one word

Remember, other `vi` commands are available as well, such as `cw`, which deletes the current word and starts insert mode.

Command Completion

The bash shell allows you to use the Tab key to complete commands. You need to type only part of a command. For example, to use the `ypdomainname` command to find the NIS domain name for your system, type the following letters:

```
# ypd
```

When you press the Tab key, bash completes the command for you. If there's more than one available command that starts with `ypd`, press the Tab key again, and you'll see a list of these commands.

Configuring the Shell

There are two sets of configuration files for any shell. Some are systemwide; in other words, they affect all users on your Linux computer. Others are user specific and are stored in a user's home directory.

Depending on your distribution, there are two basic systemwide configuration files for bash: `/etc/bashrc` and `/etc/profile`. Each of these files contains two different kinds of variables: *shell variables*, which remain constant only within a specific shell such as bash, and *environment variables*, which stay with you even if you change shells.

In other words, shell variables are local, and environment variables are global.

Shell Variables

The default Red Hat Enterprise Linux `/etc/bashrc` configuration file, shown in Figure 8.2, sets two basic shell variables: a default value for `umask` and the prompt you see with the cursor at the command-line interface.

Figure 8.2
/etc/bashrc

```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ "`id -gn`" = "`id -un`" -a `id -u` -gt 99 ]; then
    umask 002
else
    umask 022
fi

# are we an interactive shell?
if [ "$PS1" ]; then
    if [ -x /usr/bin/tput ]; then
      if [ "x`tput kbs`" != "x" ]; then # We can't do this with "dumb" terminal
        stty erase `tput kbs`
      elif [ -x /usr/bin/wc ]; then
        if [ "`tput kbs|wc -c `" -gt 0 ]; then # We can't do this with "dumb" te
rminal
          stty erase `tput kbs`
        fi
      fi
    fi
    case $TERM in
    xterm*)
        if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
            PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
```

These configuration files work with customizable files in each user's home directory. By default, they include `.bash_history`, `.bash_logout`, `.bash_profile`, and `.bashrc`. While you can customize each of these files, they contain several defaults. The periods in front of each of these files hides them from normal searches. You can view hidden files with the `ls -a` command.

.bash_history Includes a history of your previous bash commands. Some administrators do not like this file, because crackers may be able to get clues to your system from the commands you used.

You can discontinue this process by adding `HISTFILESIZE=0` to your `.bash_profile` file, as shown in Figure 8.3.

.bash_logout Sets commands for when you exit a shell. By default, this includes the `clear` command, which wipes out your previous commands from the current terminal window. A sample of the simple default `.bash_logout` file is shown in Figure 8.4.

.bash_profile Calls the `.bashrc` file for more configuration data. Adds the `~/bin` directory to your `PATH`, as shown in Figure 8.5. If you add the `HISTFILESIZE=0` variable, remember to add it to the export list in this file.

FIGURE 8.3
You can find a lot of previous commands in your *.bash_history*.

```
startx
ifconfig eth0 down
reboot
ls book
exit
startx
ls book
umount book
umount book
service smb stop
umount book
service smb start
umount book
chkconfig ntp status
chkconfig --list ntpd
chkconfig --level 35 ntpd off
chkconfig --list ntpd
ifconfig
dhclient
ifconfig
ping 207.217.126.81
ifconfig eth0 down
ifconfig eth0 up
".bash_history" 532L, 7834C                              26,4          4%
```

FIGURE 8.4
A home directory *.bash_logout*

```
# ~/.bash_logout
umount /mnt/inst

clear
~
".bash_logout" 4L, 41C
```

FIGURE 8.5
A home directory *.bash_profile*

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
BASH_ENV=$HOME/.bashrc
USERNAME="root"

export USERNAME BASH_ENV PATH
```

.bashrc Calls the `/etc/bashrc` file for basic configuration data. For the root user, this file adds aliases for the `rm`, `mv`, and `cp` commands to help prevent accidental deletion of a file, as shown in Figure 8.6. As you can see, I've added a command to connect to a shared NFS directory to my own version of this file, which is currently commented out.

FIGURE 8.6
A home directory *.bashrc*

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Another command
#mount -t nfs 192.168.1.4:/mnt/inst /mnt/inst
~
```

When you run the `export` command on a shell variable, you're essentially making it into a global, or an environment, variable. Global variables are also available to your programs.

NOTE *Remember, if you don't add the desired variables in the* `.bash*` *files in your home directory, your setup will revert to the original configuration the next time you log into Linux.*

Environment Variables

There are a large number of default environment variables, which you can review with the `env` command. You set some through `/etc/profile`. These variables include colors for filenames, settings for the secure shell, and default terminal and display variables. We've listed some of the standard environment variables in Table 8.2.

TABLE 8.2: OTHER MAJOR DEFAULT ENVIRONMENT VARIABLES

VARIABLE	DESCRIPTION
SHELL	The default shell.
LANG	The default language.
BASH_ENV	Environment variables for the bash shell, normally in `~/.bashrc`.
DISPLAY	The console used for the X Window system. DISPLAY=`:0.0` corresponds to console F7; DISPLAY=`:1` corresponds to F8; DISPLAY=`server`:0 sends GUI applications to a remote computer.
COLORTERM	The default terminal in a GUI, normally `gnome-terminal`.
PATH	Linux automatically searches through all directories in your path for a desired command, in the order shown from the output to the `echo $PATH` command. `/etc/profile` automatically adds several directories to the root user's PATH.
USER	Automatically set to the username of the currently logged-in user.

Continued on next page

Table 8.2: Other Major Default Environment Variables *(continued)*

Variable	Description
`LOGNAME`	Normally set to `$USER`.
`MAIL`	Set to the standard mail directory for a specific `$USER`.
`HOSTNAME`	Set to the output of the `/bin/hostname` command. Doesn't necessarily match `/etc/hosts` or `/etc/sysconfig/network`.
`HISTSIZE`	Sets the number of commands remembered by the history command.
`INPUTRC`	Sets defaults for keyboard mapping. See `/etc/inputrc` for details.

It's easy to reset environment variables. One of the most important of these is the `PATH`. Say you've added a number of scripts to the `/opt/data/db/programs` directory and don't want to cite the full directory path every time you want to run one of these programs. The following command adds this directory to your `PATH`:

```
# PATH=$PATH:/opt/data/db/programs
```

Now if you want to run a program such as `/opt/data/db/programs/script1`, all you need to do is type `script1` and press Enter. But remember, to make the change permanent you'll need to revise the `.bash_profile` configuration file in your home directory to reflect the change to your `PATH`. To find the current directories in your `PATH`, run the `echo $PATH` command.

Discovering the Secrets of the Shell

You can use a number of techniques with the bash shell. For example, you can direct the output of one command to a file or even to another command. The shell enables you to set up aliases to define the commands of your choice. You can also move a running program to the background, which saves you the trouble of opening another virtual terminal or console.

The bash shell is flexible; there are different ways to manage input to bash commands. For example, two different kinds of *wildcard* characters help you represent more than one file. And Linux allows you to use three kinds of quote characters to manage the input to a command.

Other secrets allow you to easily move to any home directory, set aliases that can ease administration, and move up and down the Linux directory tree quickly.

Data Streams

Linux includes three data streams: data goes in, data comes out, and errors go out a different direction. These concepts are also known as standard input (*stdin*), standard output (*stdout*), and standard error (*stderr*). Standard input normally comes from a keyboard entry to a command. For example, if you run the `ls c*` command, `c*` is standard input to the `ls` command.

Standard output is the result of a command. For example, the files you see after typing `ls` are standard output (stdout), which is normally directed to your monitor.

If there's no standard output, there may be an error message. This is the standard error data stream, which is also normally directed to your monitor.

There are two basic ways to redirect stdin, stdout, and stderr. You can pipe one of these data streams to another command, or you can redirect one of these streams to or from a file.

Redirecting Input and Output

Normally, standard input comes from a keyboard. But if you already have a file full of data, you don't need to type everything again—you can simply redirect that file of data with the left-facing arrow (<) to your program. For example, the following command directs the `database_data` file to the `database_program`:

```
# database_program < database_data
```

In many cases, you'll want to save standard output in a file. For example, the following command using the right-facing arrow (>) saves standard output from the `ls` command to the file named `filelist`:

```
# ls > filelist
```

This overwrites any data in the file named `filelist`. Alternatively, a double right-facing arrow (>>) appends data to the end of the file named `filelist`, like so:

```
# ls >> filelist
```

You can combine these redirection arrows in the same command. For example, if the `database_program` generates a lot of output, you can save it for later analysis, like so:

```
# database_program < database_data > database_output
```

Standard error output can help you diagnose trouble with a program. For example, if you have a program that runs in the middle of the night, you may want to redirect the standard error stream from this program to a file so you'll have some clues if something goes wrong. For example, the following command redirects errors to a file named `errorlog`:

```
# database_program < database_data 2> errorlog
```

Similarly, you can ensure that the previous contents of the `errorlog` file are not overwritten by using the following command:

```
# database_program < database_data 2>> errorlog
```

Tip *When you look at standard errors, be careful with the* `2>` *or the* `2>>`*. No space is allowed between these characters.*

Input and Output Pipes

Just as you can redirect stdin, stdout, and stderr to and from specific files, you can also *pipe* these data streams to other commands. If you want to review permissions on a large number of files, you can use the following two different commands:

```
# ls -l > tempfiles
# more tempfiles
```

File Descriptors and Data Streams

This sidebar is for the programmers. When a process in the shell works with a file, it sets up a file descriptor. These are program system calls, which help manage that process.

There are three standard file descriptors: `0`, `1`, and `2`. File descriptor `0` corresponds to standard input, or the right-facing arrow (>). File descriptor `1` corresponds to standard output, or the left-facing arrow (<). File descriptor `2` corresponds to standard error, which is represented by `2>` in the bash shell.

The first command takes your current file listing and stores the result in a file named `tempfiles`. The second command allows you to read the `tempfiles` file, one screen at a time. Because your file list probably changes frequently, you should delete the `tempfiles` file as it becomes out-of-date.

But this is also inefficient. You can combine these commands with a pipe (`|`), which is the character above the backslash on a U.S. keyboard. For example, the following command does the work of the previous two:

```
# ls -l | more
```

The pipe (`|`) takes the standard output from the `ls -l` command and sends the results as standard input to the `more` command. You don't need to create or delete any temporary files.

Running in the Background

Linux is a multitasking system. When you don't have additional terminals or virtual consoles available, you can still run multiple programs from a single command line. For example, some of the steps in compiling a kernel can take nearly an hour. When you run that program in the background, you don't have to wait to run other programs.

There are two ways to make programs run in the background. For example, assume you have a script in your current directory named `test`. This script starts an alarm in an hour. You want to run `test`, but you want to keep working while you wait. To do this, you can run the following command:

```
# ./test &
```

The ampersand (`&`) sends program execution to the "background." The program continues to run, and you're returned to the command-line interface.

Note *The `./programname` command runs programs in the current directory. It's the easiest way to run local programs if the current directory isn't in your `PATH`.*

Alternatively, if you're running a program to calculate the value of pi to an infinite number of digits, such a program may take a while to complete. If you forget to use the ampersand (`&`), you'll need another way to send the program to the background. The Ctrl+`Z` command suspends a running program; the `bg` command then sends the program to the background. You can return the program to the foreground with the `fg` command.

Special Shell Characters

Special shell characters regulate standard output. You may already have some special characters assigned to you in your shell. To check, run the `stty -a` command, which leads to output similar to Figure 8.7.

FIGURE 8.7 Special shell characters

```
[root@Enterprise3 root]# stty -a
speed 38400 baud; rows 12; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = M-^?; eol2 = M-^?;
start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V;
flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
-ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc ixany imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
[root@Enterprise3 root]#
```

The output shows a number of special characters and settings. In the output, the carat (^) corresponds to the Ctrl key on your keyboard. For example, the `intr = ^C` setting means that the Ctrl+`c` command interrupts a running program. Table 8.3 describes some of the default special characters. These are only defaults; you can customize the special characters you use for different commands.

TIP *The way you type a shell character varies from what you see in the output from the* `stty -a` *command. For example, the* `eof` *character appears as* `^D` *(uppercase* D*); you actually run the Ctrl+*`d` *command (lowercase* d*) to exit from the terminal.*

TABLE 8.3: SPECIAL SHELL CHARACTERS

CHARACTER	DESCRIPTION
`^C`	Interrupts and stops a running program
`^\`	Sends the `quit` command
`^D`	Stops standard input and exits from a console
`^Z`	Suspends a currently running program

There are also a number of settings with and without a hyphen in front. For example, while an `igncr` setting would ignore a carriage return, the `-igncr` setting corresponds to "Don't ignore a carriage return." In other words, when you press Enter on your keyboard, the shell gives you a new prompt. The `echo` setting means that what you type on your keyboard is seen in the terminal.

You can assign different sets of special characters with the `stty` command. For example, to suspend a program with Ctrl+`x` (instead of Ctrl+`z`), run the following command:

```
# stty susp ^X
```

WARNING *The* `stty` *command can be dangerous. For example, if you were to enter the* `stty -echo` *command, anything you typed later would not be shown on the screen. You'd have to enter the* `stty echo` *command to restore your original configuration. Imagine the frustration if a cracker were to enter* `stty -echo` *in a login profile!*

Tildes and Home Directories

One key character in the bash shell is the tilde (~). It represents the home directory of the currently logged-on user. On most standard U.S. keyboards, you can find this character on the same key as the back quote (`), just above the Tab key.

You can use the tilde with most bash shell commands. For example, users can navigate to their own home directories with the `cd ~` command. Alternatively, users can list the files in their home directories with the `ls ~` command. Other examples are shown in Table 8.4.

TABLE 8.4: COMMAND EXAMPLES WITH THE TILDE (~)

COMMAND	RESULT
`cd ~`	Navigates to your home directory.
`cd ~/.kde`	Moves to the `.kde` subdirectory of your home directory. For example, if your username is mj, this moves you to the `/home/mj/.kde` directory.
`ls ~`	Lists the files in your home directory.
`tar czvf homebk.tar.gz ~`	Backs up the files in your home directory.
`~/yourprogram`	Runs the program named `yourprogram` in your home directory.

This also can be useful in your Linux scripts, as the tilde (~) can help you configure a script to be useful for all users on your Linux server.

Connecting the Dots

The dot is nearly as important of a tool as the slash in the bash shell. While a single dot (`.`) represents the current directory, a double dot (`..`) can help you navigate to the parent directory.

You can use these dots with many bash commands. For example, the `ls .` command lists the files in the current directory, and the `ls ..` command lists the files in the parent directory.

You can even use the dot to run programs in the current directory. For example, if you're in the `/etc/rc.d/init.d` directory with the service scripts, you may not want to enter the full directory path for every command. For example, you could run the `./iptables status` command to check the current situation with your firewall.

Wildcards

There are two other special characters in Linux commands, which are variations on the Microsoft concept of wildcards. The characters are the asterisk (*) and the question mark (?). The asterisk represents any number of numbers or letters. Each question mark represents one alphanumeric character. For example, if you were to run the following command, you'd get a list of all files that start with the letter a:

```
# ls a*
```

If you have a file named `a`, it would be part of this list. In contrast, if you were to run the following command, you'd get a list of all files with two alphanumeric characters starting with `a`:

```
# ls a?
```

If you have a file named `a`, it would not be a part of this list. However, the files named `ab`, `ac`, and `ad` would. You can also perform more complex file searches with commands such as the following:

```
# ls ?at?
```

This command returns files with names such as `cate`, `kata`, and `mate`. It would not return files with names such as `Catherine`, `matador`, or `cat`.

You can even define special characters in more detail with brackets (`[]`). For example, if you want to see all files in your directory between `f0801.tif` and `f0806.tif`, you can run either of the following commands:

```
# ls f080[1-6].tif
# ls f080[123456].tif
```

TIP *In the world of Linux, the techniques associated with using wildcards are also known as* globbing.

Slashes in the Shell

There are forward slashes (`/`) and backslashes (`\`). A single forward slash represents the root directory. Additional forward slashes, such as those in `/etc/rc.d/init.d`, help you navigate to subdirectories.

The backslash is a special character. For example, if you wanted to look for an asterisk (`*`) in your `/etc/shadow` file, you could try the following command:

```
# grep * /etc/shadow
```

Unfortunately, this command looks for the name of every file in the current directory in the `/etc/shadow` file.

The problem is that the asterisk is a wildcard, which looks for almost everything, depending on the context. That's where the backslash can help. When you put the backslash in front of a special character, it "escapes" the meaning of that character.

In other words, the following command actually looks for asterisks in the `/etc/shadow` file:

```
# grep \* /etc/shadow
```

The backslash is handy for other situations, such as listing two-word directories such as Microsoft's `My Documents`. For example, if you've mounted a Microsoft Windows drive from a remote computer on the `/mnt/win1` directory, you could try to list the files in the directory with the following command:

```
# ls /mnt/win1/My Documents
```

This command looks for two separate directories: `/mnt/win1/My` and `Documents`. The problem is the space between the two words *My* and *Documents*. But when you add a backslash, the shell ignores the space and returns the list of files in the mounted `My Documents` directory, like so:

```
# ls /mnt/win1/My\ Documents
```

Quotes

There are three types of quote characters on your keyboard: the single quote ('), the double quote ("), and the back quote (`).When applied to standard input, they perform different functions.

The difference between these characters is in how they affect variables, such as `$NAME`, and shell commands, such as `date`. With any pair of quotes, the shell sends everything inside the quotes to the command. The following example uses the `echo` command. In detail, the differences are as follows:

Single quotes The shell does not process any variables or commands.

Double quotes The shell processes variables, such as `$NAME`, but does not process any commands.

Back quotes The shell tries to process every word in quotes as a command. If there are variables, they are evaluated first, and then processed as a command. Thus, if `$NAME` were in back quotes, it is processed, and then the result is evaluated as a command.

NOTE *The back quote is listed in some books as a* back tick*. I refer to the ` as a back quote, as it falls under the same category as single and double quotes in a command.*

You can see how this works in the following examples. Assume `NAME=Michael`. Remember, `date` is a command that returns the current date and time. The first command has no quotes. The shell interprets the `$NAME` variable but doesn't run the `date` command.

```
# echo Welcome $NAME, the date is date
Welcome Michael, the date is date
```

The next command encloses the input in single quotes. This prevents the shell from interpreting any variables or commands.

```
# echo 'Welcome $NAME, the date is date'
Welcome $Name, the date is date
```

The following command encloses the input in double quotes. The result is similar to the output without quotes.

```
# echo "Welcome $NAME, the date is date"
Welcome Michael, the date is date
```

The final command includes back quotes for the command. The shell interprets the command.

```
# echo "Welcome $NAME, the date is `date`"
Welcome Michael, the date is Fri Jan 17 15:52:02 EST 2003
```

Aliases

One of the most useful shell variables is the alias. When you type the `alias` command, you get a list of commands that you can substitute for others. Example aliases for the root user are shown in Figure 8.8.

FIGURE 8.8
A list of aliases

```
[root@Enterprise3 root]# alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-ti
lde'
[root@Enterprise3 root]# 
```

As discussed earlier, this list shows aliases for the `cp`, `mv`, and `rm` commands, which can help prevent the accidental deletion of a file. It's easy to create other aliases. For example, the following command makes `rx` an alias for the `redhat-config-xfree86` command:

```
# alias rx=redhat-config-xfree86
```

Now you can type the `rx` command, and the bash shell calls up the `redhat-config-xfree86` utility. You can reverse the process with the `unalias` command. For example, the following command deletes the alias for `redhat-config-xfree86`:

```
# unalias rx
```

TIP *You can create aliases for complex commands that you run frequently. For example, the* `alias le='ls -ltr /etc | more'` *command could be a timesaver when you need to look through the* `/etc` *configuration files. If you want to make the alias change permanent, add the change to the* `.bashrc` *file in your home directory.*

Creating Basic Scripts

In the following sections, we introduce the basic scripts available in Linux. You can customize some of them to meet your own needs. Some of you may have extensive experience with shell scripts; for you, these sections are trivial. They're designed for newer Linux users without scripting experience.

You've already seen a number of sample scripts, such as those in user's home directories for the bash shell. Other scripts are already configured to run on a regular basis; we address them in the section on the `cron` daemon in Chapter 13.

Basic Script Language

Linux shell scripts are predominantly written to the bash shell. You can verify this with the first line in the script. The following line is not a comment, but it tells Linux that this script conforms to the commands associated with bash:

```
#!/bin/sh
```

Once bash is established for this script, then you can add the `PATH` you need; alternatively, you can use the full directory path to the commands you want to use. For example, the following command from `/etc/crontab` sets the `PATH` for the remainder of the script:

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

You can still run the commands of your choice using the full path to all files and directories. For example, take the following command from the `logrotate` script in the `/etc/cron.daily` directory:

```
/usr/sbin/logrotate /etc/logrotate.conf
```

If you added the noted `PATH` command, the `/usr/sbin` would not be required to point to the `logrotate` command; however, the `PATH` did not include the `/etc` directory for the `logrotate` configuration file.

You can build programming constructs into a script. The most basic of these is the conditional, also known as an `if` conditional. For example, you can set up a script to run one command if a certain condition is met or another command if it is not met. You'll see it in a script in a format similar to the following:

```
if [ condition is met ]
   then run command 1
   else run command 2
fi
```

Naturally, there are a lot of ways to see if the ***`condition is met`***. I've described some of them in Table 8.5. One key operator is the exclamation point, which reverses the meaning. For example, the following command checks if `/etc/file` does not exist:

```
if [ ! -f /etc/file ]
```

Table 8.5: Shell Conditionals

Conditional	Description
`if [ x -eq y ]`	Does x=y?
`if [ x -ne y ]`	Does x≠y?
`if [ x -gt y ]`	Does x>y?
`if [ x -ge y ]`	Does x≥y?
`if [ x -lt y ]`	Does x<y?
`if [ x -le y ]`	Does x≤y?
`if [ -s /etc/file ]`	Does `/etc/file` have data?
`if [ -f /etc/file ]`	Is `/etc/file` a regular file?
`if [ -d /etc/dir ]`	Is `/etc/dir` a directory?
`if [ x -ne y -a x -lt z ]`	Does x≠y and x<z?

You'll see shortly that there's often more than one way to describe a conditional; for example, the following command also checks to see if x≠y:

```
if [ x != y ]
```

Sample Scripts

Let's continue looking at the `logrotate` script. It's fairly simple. As described earlier, it runs the `logrotate` command, based on the `/etc/logrotate.conf` configuration file. If the command is successful, it returns a value of 0, which is given to the `EXITVALUE` variable.

```
EXITVALUE=$?
```

Now this script evaluates the value of `EXITVALUE`. The following command checks to see if it is not equal to 0:

```
if [ EXITVALUE !=0 ]; then
```

An `EXITVALUE` other than 0 indicates that the `logrotate` command wasn't successful. And the following command adds a message to that effect to your `/var/log/messages` file:

```
/usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
```

The commands that follow end the `if` loop and then exit the script.

```
fi
exit 0
```

Create Your Own Script

If you've never created a script before, the description of the `cron` daemon scripts in Chapter 13 should be instructive. Those scripts run automatically on a schedule. Sometimes, you'll want to run scripts on an "as-needed" basis. They can be simple; for example, you could include the following command in a simple script that you can use to mount the network installation source as needed:

```
mount -t nfs server.example.com:/mnt/inst /mnt/inst
```

All you need to do is save it in a text file in your `~/bin` directory and make the script executable. The `~/bin` directory is a subdirectory of your home directory and is part of the default `PATH` for every regular user. For example, if you saved this in your `~/bin/inst` text file, all you need to do is make it executable; then you could run this script with the `inst` command.

Make It Executable

Making a script executable is a matter of changing permissions, and possibly ownership. I discussed both concepts in Chapter 6. Assuming you created the script you wanted to use while logged into your account, ownership shouldn't be a problem. However, if you created a script as root and want to use it as a regular user, you may need to change ownership. For example, based on the `~/bin/inst` script described earlier, you could run the following command to give ownership to user michael:

```
# chown michael.michael ~/bin/inst
```

Changing permissions also requires a straightforward command. The following command makes the script executable for the owner of the file:

```
# chmod 744 ~/bin/inst
```

You can now run the `inst` command to execute the script you configured in the `~/bin/inst` file.

Summary

While the bash shell includes a large number of commands, the details of the bash shell are fairly straightforward. Interactivity makes it easy to recall previous commands. Command completion allows you to find the command that you need with just a couple of strokes of the Tab key.

Shell and environment variables are maintained in some basic configuration files in the `/etc` and individual users' home directories. These variables determine the basic setup of the command-line interface. Other variables determine the size of your history, the default terminal, standard e-mail directories, and more.

There are a number of secrets associated with the shell. They include the three basic data streams: standard input, standard output, and standard error. Also, you can run commands in the background. Special shell characters set terminal parameters and allow you to use the Ctrl key to perform different tasks on your keyboard. The tilde (~) represents any user's home directory.

Dots and double dots can help you navigate through the Linux filesystem hierarchy. Wildcards help you identify files and commands even when you don't know the complete name. While forward slashes help you navigate directories, backslashes escape the meaning of characters such as asterisks and spaces. Single, double, and back quotes let you process variables and commands in different ways. Aliases make it possible to rename commands you may otherwise forget.

Scripts are a collection of shell commands in an executable file. You can configure them in your `PATH` in the `~/bin` directory. Don't forget to change their permissions to make them executable. I'll describe a number of preconfigured scripts in future chapters; for example, I show you how `cron` daemon scripts work in Chapter 13.

That completes Part II, where we have examined the fundamentals of Linux. In Part III, we'll look at a number of basic administrative functions. Chapter 9 begins this process by showing how you administer users and groups. Red Hat Enterprise Linux promotes security by helping you administer users and groups with the Shadow Password Suite, quotas, and the User Private Group scheme.

Part 3

Basic Linux Administration

In this Part, you will learn:

Chapter 9

Administering Users and Groups Securely

ONE OF THE KEY tasks for a Linux administrator is to maintain users and groups. Even if the computer is a workstation with one dedicated user, chances are that you'll want to maintain at least a root and a regular account on that computer.

You configure users and groups in several basic files in the `/etc` directory. Red Hat Enterprise Linux allows you to create users by editing these files directly, or you can use some basic commands or even run the Red Hat User Manager. In any case, there are other commands and configuration files that you can use to manage the life of a user account and the associated password. With `/etc/sudoers`, you can assign administrative privileges to the users of your choice. You can even limit the right to use `su` to users in the wheel group.

The Shadow Password Suite allows Linux to provide an additional layer of protection, by user and by group. Quotas can help you regulate the amount of space and/or the number of files that users are allowed on your system.

Red Hat Enterprise Linux includes a different way of organizing groups, known as the User Private Group scheme. It enhances user security, because users get their own exclusive group. If needed, it still allows you to set up individual groups with a shared directory. This chapter covers the following topics:

- Basic user and group management
- Creating users
- Using the Shadow Password Suite
- Setting quotas
- Creating user private groups

Basic User and Group Management

Everyone on a Linux system needs a user account. Every account has rights and privileges that vary depending on the command and the directory. Linux user accounts are organized into groups. While default users are the only member of their default groups, you can organize users into new groups, and you can configure rights and privileges that vary differently by group.

In Red Hat Enterprise Linux, user accounts are organized in `/etc/passwd`. Their passwords are made more secure in `/etc/shadow`. For Red Hat Enterprise Linux groups, the analogous files are `/etc/group` and `/etc/gshadow`.

When creating a new account, the default parameters are configured in `/etc/login.defs`; configuration files are normally copied to the new user's home directory from the `/etc/skel` directory.

NOTE *Regular users will want their own accounts, and generally, you want to minimize risks by keeping them away from root user privileges. However, if you're the administrator for your Linux computer, you may want to sign in as the root user, for the reasons discussed in Chapter 6.*

/etc/passwd

Linux users can be classified into three groups: administrative, service, and regular users. Every user has rights and privileges. Regular and administrative users have usernames, passwords, and home directories. All users are configured through a line in the `/etc/passwd` file, as shown in Figure 9.1.

The last nine lines in this figure contain entries for regular users. As you can see, usernames are associated with services such as `ftp`, `apache`, and `squid`. Each entry includes seven columns delineated by colons (:). Table 9.1 describes each of these columns.

TABLE 9.1: */ETC/PASSWD* ENTRIES

COLUMN	FUNCTION	COMMENT
1	Username	Login name.
2	Password	If this field contains an `x`, the encrypted password is stored in `/etc/shadow`.
3	User ID	Red Hat user IDs start at 500.
4	Group ID	Red Hat group IDs normally match user IDs.
5	Extra information	Commonly used for a user's real name.
6	Home directory	Normally `/home/`*`username`*.
7	Default shell	The shell a user sees after logging in.

/etc/shadow

Red Hat Enterprise Linux includes the `/etc/shadow` file for additional password security. By default, this file is readable only to the root user. If you use standard commands to create new users, basic information is also added to this file, based on the defaults in `/etc/login.defs` (which we discuss later in this chapter). Take a look at `/etc/shadow` in Figure 9.2.

FIGURE 9.1
/etc/passwd

```
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
gdm:x:42:42::/var/gdm:/sbin/nologin
desktop:x:80:80:desktop:/var/lib/menu/kde:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
netdump:x:34:34:Network Crash Dump user:/var/crash:/bin/bash
pvm:x:24:24::/usr/share/pvm3:/bin/bash
vp:x:501:501:Vladimir Putin:/home/vp:/bin/bash
ez:x:502:502::/home/ez:/bin/bash
mj:x:500:500:Mike1:/home/mj:/bin/bash
michael:x:504:504::/home/michael:/bin/bash
donna:x:505:505::/home/donna:/bin/bash
elizabeth:x:506:506::/home/elizabeth:/bin/bash
nancy:x:507:507::/home/nancy:/bin/bash
randy:x:508:508::/home/randy:/bin/bash
mt:x:509:509:Mike2:/home/mt:/bin/bash
                                                              24,1          95%
```

FIGURE 9.2
/etc/shadow

```
mailnull:!!:12348:0:99999:7:::
smmsp:!!:12348:0:99999:7:::
pcap:!!:12348:0:99999:7:::
xfs:!!:12348:0:99999:7:::
ntp:!!:12348:0:99999:7:::
gdm:!!:12348:0:99999:7:::
desktop:!!:12348:0:99999:7:::
apache:!!:12348:0:99999:7:::
webalizer:!!:12348:0:99999:7:::
squid:!!:12348:0:99999:7:::
postfix:!!:12348:0:99999:7:::
named:!!:12348:0:99999:7:::
netdump:!!:12348:0:99999:7:::
pvm:!!:12361::::::
vp:$1$ITOduHVt$JL8ZAJWEG2IONZklVN7yX/:12363:0:99999:7:::
ez:$1$uIdcBpTj$zJCykpZDoGJGg8qEwpdEn0:12373:0:99999:7:::
mj:$1$72ilkbjP$mpCtqfk7wac67t9mKDVRC/:12402:0:99999:7:::
michael:$1$bxeI4qn.$Mu1Gj34B8TFyb5mde9vg51:12377:0:99999:7:::
donna:$1$skAC6epj$Qls.oh2UCMO4eFhAS18Jp/:12376:0:99999:7:::
elizabeth:!!:12373:0:99999:7:::
nancy:!!:12373:0:99999:7:::
randy:!!:12373:0:99999:7:::
mt:!!:12381:0:99999:7:::
                                                              24,1          95%
```

As you can see, the last lines contain entries for the same regular and service users that were shown in `/etc/passwd`. In this case, each user entry includes eight columns delineated by colons (:). Table 9.2 describes each of these columns.

TABLE 9.2: */ETC/SHADOW* **ENTRIES**

COLUMN	FUNCTION	COMMENT
1	Username	Login name.
2	Password	Encrypted password.
3	Number of days	Last time the password was changed, in days, after January 1, 1970.
4	Minimum password life	You can't change a password for at least this amount of time, in days.

Continued on next page

Table 9.2: */etc/shadow* Entries *(continued)*

Column	Function	Comment
5	Maximum password life	You have to change a password after this period of time, in days.
6	Warning period	You get a warning this many days before your password expires.
7	Disable account	If you don't use your account this many days after your password expires, you can't log in.
8	Account expiration	If you don't use your account by this date, you won't be able to log in. May be in YYYY-MM-DD format or in the number of days after January 1, 1970.

/etc/group

The Red Hat Enterprise Linux group configuration file is simpler than those for users; they include only four columns. In Figure 9.3, you can see the same regular usernames in `/etc/group` that you saw in `/etc/passwd` and `/etc/shadow`. In `/etc/group`, they are group names. You may note that the group ID for groups such as mj and vp matches the user IDs for the users with the same names in the previous two configuration files. You may also note additional groups with higher user IDs, which you can use for groups of multiple users.

Note the final entry, the angels group. As you can see, users nancy and randy are members of that group. Table 9.3 describes the columns in `/etc/group`.

Table 9.3: */etc/group* Entries

Column	Function	Comment
1	Group name	By default, Red Hat users are members of groups with the same name.
2	Password	If you see an `x` in this column, see `/etc/gshadow` for the actual encrypted password.
3	Group ID	By default, Red Hat users have the same ID as their groups.
4	Members	Includes the usernames of others who are members of the same group.

/etc/gshadow

The Red Hat Enterprise Linux `/etc/gshadow` configuration file for groups is analogous to the `/etc/shadow` file for users. It specifies an encrypted password for applicable groups, as well as administrators with privileges for a specific group. Figure 9.4 shows a sample `/etc/gshadow` file.

FIGURE 9.3
/etc/group

```
ntp:x:38:
gdm:x:42:
desktop:x:80:
apache:x:48:
webalizer:x:67:
squid:x:23:
postdrop:x:90:
postfix:x:89:
named:x:25:
netdump:x:34:
pvm:x:24:
vp:x:501:
ez:x:502:
managers:x:1000:ez
project:x:1001:vp
mj:x:503:
michael:x:504:
donna:x:505:
elizabeth:x:506:
nancy:x:507:
randy:x:508:
mt:x:509:
angels:x:2000:nancy,randy
                                                  58,9          Bot
```

FIGURE 9.4
/etc/gshadow

```
xfs:x::
ntp:x::
gdm:x::
desktop:x::
apache:x::
webalizer:x::
squid:x::
postdrop:x::
postfix:x::
named:x::
netdump:x::
pvm:!::
vp:!!::
ez:!::
managers:!!::ez
project:!!::vp
mj:!::
michael:!::
donna:!::
elizabeth:!::
nancy:!::
randy:!::
mt:!::
                                                  45,1          97%
```

Note the differences from `/etc/group` with respect to the sharing group. Table 9.4 describes the columns in `/etc/shadow`.

TABLE 9.4: */ETC/SHADOW* ENTRIES

COLUMN	FUNCTION	COMMENT
1	Group name	You can create additional groups.
2	Password	The encrypted group password, added with the `gpasswd` command.
3	Group administrator	The user allowed to manage users in that group.
4	Group members	Includes the usernames that are members of the same group.

/etc/skel

Users have a default set of configuration files and directories. You examined some of these files as they related to the bash shell in Chapter 8. The default list of these files is located in the `/etc/skel` directory, which you can easily inspect with the `ls -la /etc/skel` command, as shown in Figure 9.5. The list changes depending on what you have installed.

FIGURE 9.5
Default home files in */etc/skel*

```
[root@Enterprise3 root]# ls -la /etc/skel/
total 40
drwxr-xr-x    3 root     root         4096 Oct 23 15:57 .
drwxr-xr-x   67 root     root         8192 Mar  4 21:57 ..
-rw-r--r--    1 root     root           24 Sep 18 08:26 .bash_logout
-rw-r--r--    1 root     root          191 Sep 18 08:26 .bash_profile
-rw-r--r--    1 root     root          124 Sep 18 08:26 .bashrc
-rw-r--r--    1 root     root          237 May 22  2003 .emacs
-rw-r--r--    1 root     root          120 Aug 20  2003 .gtkrc
drwxr-xr-x    3 root     root         4096 Oct 23 15:40 .kde
-rw-r--r--    1 root     root          220 Nov 28  2002 .zshrc
[root@Enterprise3 root]#
```

TIP *If you have a list of standard files, such as corporate policies for new users, you may want to copy them to* `/etc/skel`. *All new users will get a copy of these files in their home directories.*

/etc/login.defs

When you create a new user, the basic parameters come from the `/etc/login.defs` configuration file. The version included with Red Hat Enterprise Linux includes settings for e-mail directories, password aging, user ID numbers, and group ID numbers and for creating a home directory. The following default variables in this file are almost self-explanatory:

```
MAIL_DIR   /var/spool/mail   # Default mail directory
PASS_MAX_DAYS 99999          # Password max life
PASS_MIN_DAYS 0              # Password min life
PASS_MIN_LEN 5               # Min password length
PASS_WARN_AGE 7              # Warning before expiration
UID_MIN      500             # Lowest User ID number
UID_MAX    60000             # Highest User ID number
GID_MIN      500             # Lowest Group ID number
GID_MAX    60000             # Highest Group ID number
CREATE_HOME                  yes
```

Needless to say, these settings can be further refined through other configuration files. For example, you can manage the allowed lifetime settings for passwords by editing `/etc/shadow`. You can review a copy of this file in Figure 9.6.

FIGURE 9.6
/etc/login.defs

```
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#QMAIL_DIR      Maildir
MAIL_DIR        /var/spool/mail
#MAIL_FILE      .mail

# Password aging controls:
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7

# Min/max values for automatic uid selection in useradd
UID_MIN                   500
UID_MAX                 60000
# Min/max values for automatic gid selection in groupadd
GID_MIN                   500
GID_MAX                 60000

# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#USERDEL_CMD    /usr/sbin/userdel_local

# If useradd should create home directories for users by default
# On RH systems, we do. This option is ORed with the -m flag on
# useradd command line.
CREATE_HOME     yes
                                                         22,1          Top
```

Administering User Accounts

Linux administrators perform three basic tasks with user accounts. They add new users. They delete users. They manage the access parameters of existing users. While a Red Hat Enterprise Linux graphical tool is available for this purpose, most administrators perform these functions from the command-line interface.

Adding Users

The following are the three basic ways to add users in Red Hat Enterprise Linux:

- Edit the `/etc/passwd` file directly, adding desired files to new users' home directories.
- Work with some of the commands designed for this purpose, such as `useradd`.
- Open the graphical front end, the Red Hat User Manager (`redhat-config-users`).

Alternatively, the `newusers` command lets you add a whole group of users based on a batch file configured to the same format as `/etc/passwd`. The limitation to Linux usernames is that they can't start with a number or an uppercase letter.

The Command Line vs. GUI Administrative Debate

Linux administrators generally prefer tools at the command-line interface. While this may appear archaic to a Microsoft Windows administrator, there are good reasons to use command-line tools:

- Command-line tools are more versatile. Generally, more options are available when you use a command-line tool than when you use a GUI.
- Command-line tools are faster. You don't have to wait for Linux to process the GUI or to place another GUI tool on your screen.
- Command-line tools are easier to run from remote computers. Because you don't have the overhead of the GUI, it's easier to administer remote computers from a variety of terminals. This can be a terrific advantage in the enterprise.
- GUI tools are just front ends. In other words, Linux GUI tools take the entries you make and run the corresponding command in the shell.
- GUI tools are another layer of software—which is another area where things can go wrong.
- GUI tools don't show all errors. While command-line interface tools give you error messages that you can see at the console, GUI tools may not show these errors on a graphical desktop.

The Direct Method

It's instructive to go through the steps required to create a new user . It can help you appreciate all of the parameters associated with existing users. For this example, assume you're creating an account for James K. Polk (U.S. president, 1845–1849), and plan to assign him user ID and group ID 600. (If 600 is already taken, substitute a different unused number between 500 and 60000.) Follow these steps to set up the user account:

1. Open `/etc/passwd` in a text editor.
2. Start a new line. The easiest way to do this is by copying the applicable information from a current user.
3. Change the username, user ID, group ID, and home directory. Insert **`jkp`** as the username in the first column, **`600`** in the third column for the user ID, **`600`** in the fourth column for the group ID, **`James K Polk`** in the fifth column, and **`/home/jkp`** as the user's home directory in the sixth column. Make sure that the information you enter (except for the shell) is unique relative to other entries in your `/etc/passwd` file. Save your changes to this file.
4. Open `/etc/shadow` in a text editor. Create a new line by copying the applicable information from a current user. Insert **`jkp`** for the new user in the first column. Save your changes to this file. This is a read-only file; in `vi`, the `:wq!` command overrides read-only settings if you own the file.
5. Open `/etc/group` in a text editor. Create a new line by copying the applicable information from a current group. For this user, insert **`jkp`** as the group name in the first column and **`600`** as the group ID in the third column.

6. Set up your new user's home directory. For user jkp, the appropriate command is as follows:

```
mkdir -p /home/jkp
```

7. Give your new user access to his home directory. In this case, assign ownership with these commands:

```
chown jkp /home/jkp
chgrp jkp /home/jkp
```

8. Assign a new password with the `passwd jkp` command. Give the new password to your new user. Tell him to assign a new password to himself with the `passwd` command.
9. Copy the basic initialization files, which are normally stored in the `/etc/skel` directory. (We covered these files, such as `.bashrc` and `.bash_profile`, in Chapter 8.) Change your identity to the new user with the following command:

```
su - jkp
```

   Copy these files with this command:

```
cp /etc/skel/.* /home/jkp
```

10. Copy any subdirectories in `/etc/skel` to `/home/jkp`. For example, you can copy the `/etc/skel/.kde` directory with the following command:

```
cp -r /etc/skel/.kde /home/jkp
```

11. Change the user and group ownership of the files that you copied from `/etc/skel`.
12. Log out from the jkp user account, and tell your new user about his new username and password.
13. Assuming you're using the default Shadow Password Suite, run the `pwconv` and `grpconv` commands. These commands are discussed later in this chapter.

Using *useradd*

It's a lot easier to use the `useradd` command to create a new user. For example, to set up a new account for jkp, all you need to do is type in the following command:

```
# useradd jkp
```

This command sets up user jkp with the defaults as described earlier in the `/etc/login.defs` configuration file. It also copies the files from `/etc/skel` and modifies the ownership of these files. You still need to assign a new password for jkp, as described in step 8 of the previous section.

Inspect your `/etc/passwd`, `/etc/shadow`, and `/etc/group` configuration files to verify that the `useradd` command added entries for jkp to these files.

Using *newusers*

The `newusers` command can handle a large number of users from a batch file of usernames and passwords, in the same format as the `/etc/passwd` file. The only difference is that the password column requires an encrypted password, which you can copy from the `/etc/passwd` or `/etc/shadow` entry for a known user. If you create a list of new users in a file named `new-batch`, you can then set up these users with the following command:

```
# newusers new-batch
```

The `new-batch` file must be in the same format as `/etc/passwd`; the passwords must be entered in clear text. Therefore, if you save this batch file, make sure it's secure. You might want to hide it, encrypt it, or delete it. A list of usernames and clear-text passwords is a tempting tool for anyone who wants to crack your system.

Tip *It's easy to copy text such as an encrypted password. Try it yourself. Open* `/etc/shadow` *in the text editor of your choice. Highlight the password. Exit from the editor. Right-click your mouse. You should see an exact copy of what you just highlighted. Open the file of your choice. In insert mode, when you right-click your mouse again, you should see another copy of the encrypted password.*

Deleting Users

You can delete users directly, or you can use the `userdel` command. You can even deactivate a user temporarily while retaining the files in that user's home directory.

Tip *It's easy to deactivate a user. Just substitute an asterisk (***) for the target user's password in* `/etc/passwd`*. That user won't be able to log in to her account with any password. This works even if you're using the default Red Hat Enterprise Linux Shadow Password Suite.*

The Direct Method

Deleting users is easier than adding them. You just need to ensure that the user's entries are deleted from the respective configuration files and then delete that user's home directory. The basic steps are as follows:

- Delete the user's entry from `/etc/passwd`.
- Delete the user's entry from `/etc/group`.
- Delete the user's entry from `/etc/shadow`.
- Delete the user's entry from `/etc/gshadow`.
- Delete the user's home directory after saving the files you need.

Deleting with Commands

When working with commands, two steps that are required to delete a user. The `userdel` command is straightforward. If you have a user named James K. Polk who just left your company, you'll want

to deactivate his account. Retrieve and save any files you need from his home directory, and then run the following command:

```
# userdel -r jkp
```

This command deletes jkp's information from the `/etc/passwd` file. The `-r` switch deletes the `/home/jkp` directory, including any files and directories that it may contain.

But you also need to delete that user's group with the `groupdel` command, as shown here. Otherwise, the next user you add will have a user ID and a group ID that differ from each other, which can cause problems when you manage new users in the future.

```
# groupdel jkp
```

Managing User Access with *chage*

You can manage user passwords with the `chage` command. It can help you specify the information described in the earlier discussion on `/etc/shadow`, based on regulating the lifetime of a password. In fact, `chage` changes the settings in this file. You can review the switch associated with `chage` in Table 9.5.

TABLE 9.5: *CHAGE* **COMMANDS**

COMMAND	RESULT
`chage -m` ***`days user`***	Sets the minimum life of a password to *days* days.
`chage -M` ***`days user`***	Sets the maximum life of a password to *days* days.
`chage -I` ***`days user`***	Sets the number of *days* that an account can be inactive before it's locked.
`chage -E` ***`date user`***	Sets the *date* after which an account is inaccessible.
`chage -W` ***`days user`***	Sets an advance warning, in *days*, of an upcoming required password change.
`chage -l` ***`user`***	Lists the current user's password and account information. Can be run by regular users on their own accounts.

The date can be in YYYY-MM-DD format, or in the number of days after January 1, 1970.

The Red Hat User Manager

You can use the Red Hat User Manager utility to manage the users and groups with accounts on your Linux system. Start it from a GNOME desktop by selecting Main Menu ➢ System Settings ➢ Users and Groups. Alternatively, you can start it from a GUI command-line interface with the `redhat-config-users` command. This opens the Red Hat User Manager window, shown in Figure 9.7.

As you can see, this window includes two tabs. The Users tab lists current users on the system, from `/etc/passwd`. The categories should be familiar if you know this file. To add a user, click Add User. This opens the Create New User dialog box, shown in Figure 9.8.

This dialog box allows you to enter the information associated with the new user, along with the password. Normally, the new user gets the next user ID available, in this case, 501. If you activate Specify User ID Manually, you can set the number of your choice.

You can add more account information for each user. Highlight a user and click Properties. This opens the User Properties dialog box, shown in Figure 9.9.

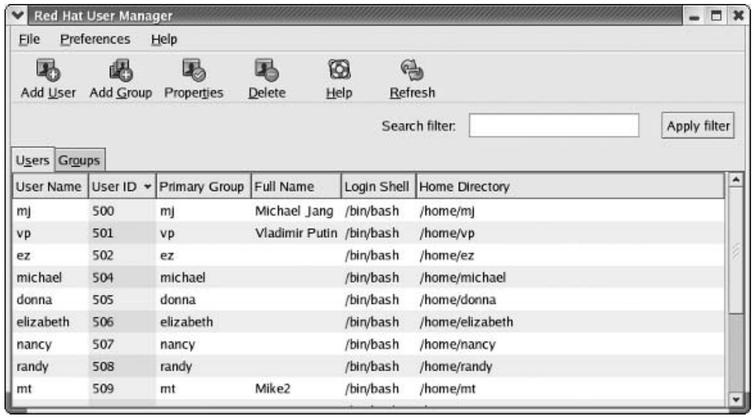


FIGURE 9.7
The Red Hat User Manager

FIGURE 9.8
Creating a new user

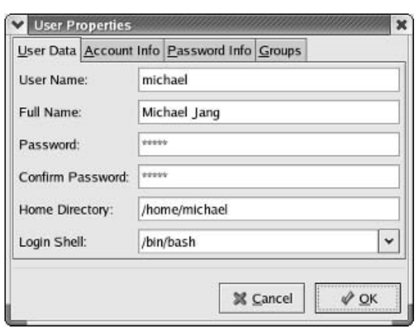


FIGURE 9.9
Changing user properties

There are four tabs of information within User Properties, which are described in Table 9.6.

TABLE 9.6: CONFIGURABLE USER PROPERTIES

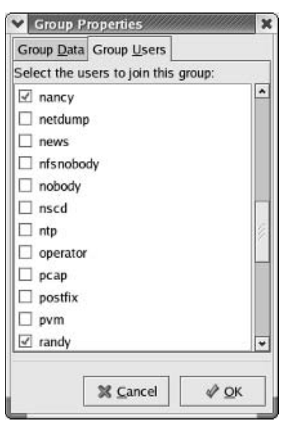
TAB	DESCRIPTION
User Data	Lists basic data for the user, stored in `/etc/passwd` and `/etc/shadow`.
Account Info	Allows you to lock and/or set an expiration date for the account; the information is stored in `/etc/shadow`.
Password Info	Lets you set up password expiration parameters; the information is stored in `/etc/shadow`.
Groups	Permits you to set group membership for that user; the information is stored in `/etc/group`.

Click OK to return to the main Red Hat User Manager window. Next, select the Groups tab, which lists current groups from `/etc/group`. Click Add Group. This opens the Create New Group dialog box, shown in Figure 9.10. By default, each user is a member of his or her own group, with the same ID number. For example, user donna has a user ID of 505, and group donna has a group ID of 505. This is the User Private Group scheme described later in this chapter.

FIGURE 9.10
Creating a new group

Whenever you create a special group, it's a good idea to assign it a group ID number in a different range from your user IDs. For example, I've created the group named *angels*. After selecting angels from the Groups tab, I clicked the Properties button, which opens the Group Properties dialog box. On the Group Users tab shown in Figure 9.11, you can add the users of your choice to this new group, in this case, nancy and randy.

FIGURE 9.11
Adding users to a group



The root Account and *sudoers*

As an administrator in the enterprise, you may want to share your responsibilities. Suppose you aren't ready to give user elizabeth full root privileges, but you do want to let her configure and manage the FTP server on your system. For this purpose, you can set up elizabeth in the `/etc/sudoers` file.

NOTE *You can't edit* `/etc/sudoers` *with the* `vi` *command. To open it for editing, you need to use the* `visudo` *command.*

In this file, you can give different users the privileges of your choice. The default command in this file is straightforward; it allows the root user to do anything from ALL computers:

```
root   ALL=(ALL) ALL
```

You can set up users in the wheel group, which we describe in the next section. If you activate the following command from the default `/etc/sudoers` file, all users who belong to the wheel group also get root user privileges:

```
# %wheel   ALL=(ALL)  ALL
```

Now let's give user elizabeth privileges for the default FTP server, vsFTP. To do this, add the following command to `/etc/sudoers`, which allows elizabeth to start and stop the `vsftpd` daemon:

```
elizabeth ALL=(root) NOPASSWD: /etc/rc.d/init.d/vsftpd
```

The `NOPASSWD:` means no verification is required. If you leave this out, elizabeth will have to reenter her own (not the root user) password. Now the next time elizabeth logs into Linux, she can start the vsFTP server with the following command:

```
$ sudo /etc/rc.d/init.d/vsftpd start
```

NOTE *Commands configured in* `/etc/sudoers` *don't work unless the configured user starts the allowed command with* `sudo`*.*

The format of commands in `/etc/sudoers` is configured in the following order, as defined in Table 9.7:

```
user host run_as command
```

TABLE 9.7: */ETC/SUDOERS* **COMMAND FORMAT**

ENTRY	DESCRIPTION
`user`	One or more users; can substitute from `/etc/group` with a %; for example, `%wheel` applies to all users in the wheel group.
`host`	One or more computer hostnames.
`run_as`	Users to run as; common options are `root` and `ALL`.
`command`	One or more root-level commands that you want the user or group to run.

You can set up a group of users, hosts, or commands as part of an alias. There are four basic alias categories: users, hosts, commands, and "run as." The first three are self-explanatory. A "run as" alias is the user whose privileges you use. This is normally `root`, a service user such as `apache`, or `ALL` for all users on that computer.

For more information on how this works, refer to the sudo website at `www.sudo.ws`. The University of Wisconsin has an interesting way of securing their Linux computers in `/etc/sudoers`. It configures different groups of users and allows commands for each group. It specifically limits access to different command shells, as well as the `su`, `passwd`, and `visudo` commands. You can review their default configuration at `post.doit.wisc.edu/linux/secure.html`.

Limiting root Access with wheel

By default, any regular user who knows the root password can run the `su` command to access root account privileges. If you want to limit root access to one or more regular users, you'll want to do two things.

First, add the users who you want to allow access to `su` to the wheel group, in `/etc/group`. Second, activate the following command in `/etc/pam.d/su`:

```
auth   required   /lib/security/$ISA/pam_wheel.so use_uid
```

This command takes advantage of Pluggable Authentication Modules, which we describe in detail in Chapter 17.

Using the Shadow Password Suite

The Shadow Password Suite features all of the commands related to managing Linux users and groups, including those already addressed in this chapter. By default, Red Hat Enterprise Linux uses this suite to provide additional security through encrypted passwords in the `/etc/shadow` and `/etc/gshadow` files. These files require commands to convert passwords to and from the companion `/etc/passwd` and `/etc/group` configuration files.

These encrypted password files have more restrictive permissions than `/etc/passwd` or `/etc/group`; only the root user is allowed to even view these files, and they are not writeable by default.

However, these additional security provisions may not do you much good unless your passwords are strong, as we explain the next section.

Note *One of the major password-testing programs is known as* crack*. A version of it is available as part of the* `cracklib*` *RPM packages. You should use it only to test the security of your users' passwords.*

Strong Passwords

By default, Red Hat Enterprise Linux discourages using simple passwords, such as dictionary words, or simple patterns, such as *abcd*. Readily available password-cracking programs can decipher such passwords in minutes. In contrast, the best passwords are based on a combination of uppercase and lowercase letters, numbers, and punctuation; such passwords can take weeks for the same programs to decipher. One easy way to set up such passwords is based on a favorite sentence; for example, Ira3mmoW could stand for "I ran a 3 minute mile on Wednesday."

NOTE *When you use the* `passwd` *command, you get to type in the new desired password twice. If your passwords don't match, you'll see a warning to that effect. After pressing Enter, you're then taken to the original prompt where you can try again.*

Converting User Passwords

Two commands are associated with converting user passwords in the Shadow Password Suite: `pwconv` and `pwunconv`.

pwconv Converts an existing `/etc/passwd` file. Passwords that currently exist in `/etc/passwd` are replaced by an `x`; the encrypted password, username, and other relevant information are transferred to the `/etc/shadow` file. If you've recently added new users by editing the `/etc/passwd` file in a text editor, you can run this command again to convert the passwords associated with any new users. This works even if other passwords are already encrypted in `/etc/shadow`.

pwunconv Passwords are transferred back to `/etc/passwd`, and the `/etc/shadow` file is deleted. Be careful, because this also deletes any password-aging information (see the `chage` command described earlier) otherwise saved in `/etc/shadow`.

Converting Group Passwords

As we discussed earlier, you can configure group administrators in `/etc/group` and assign associated passwords with `gpasswd`. Once you have group passwords, you may have the same security concerns as with regular user passwords. Two commands are associated with converting user passwords in the Shadow Password Suite: `grpconv` and `grpunconv`.

grpconv Converts an existing `/etc/group` file. The relevant information is transferred to `/etc/gshadow`.

grpunconv Reverses the process of the `grpconv` command; like `pwunconv`, this command also deletes any existing `/etc/gshadow` file.

Setting Quotas

Quotas keep individual users or groups from consuming all the space available on a partition. Linux administrators commonly use disk quotas to regulate the amount of space occupied by any single user for e-mail, website files, FTP files, and more. This prevents any particular user from uploading so much data that it crowds out a critical directory such as `/boot` or root (`/`). Without sufficient free space for these directories, Linux might even crash.

You can configure quotas with limits on the number of inodes. Each inode is associated with a specific file. Alternatively, you can set absolute limits in kilobytes. In other words, you can limit the number of files that a user or group can put on your system, or you can place an absolute limit on the amount of data that user or group can place on your system.

Quotas allow you to monitor the pattern of use of your system.

Configuration

By default, the quota RPM package is installed and is active in Red Hat Enterprise Linux. If you're not sure, run the following command:

```
rpm -q quota
```

If the packages are installed, you'll see the package name and version number in the standard output on your screen. If necessary, see Chapter 10 for instructions on how to install RPM packages such as `quota-*`.

Quotas are normally active in the kernel. Once they're active, you can configure quotas on a specific partition for users and/or for groups. In either case, you'll need to remount the target directory with active quota settings shown in `/etc/fstab`. Once you've configured those settings, you can activate quotas yourself; they are activated in subsequent reboots in `/etc/rc.d/rc.sysinit`.

Kernel Notes

While the default Red Hat Enterprise Linux kernel enables quotas, that setting may not apply for kernels you download from other sources. Fortunately, checking the appropriate kernel setting is easy.

When you download the source code for a kernel, the files are saved to the `/usr/src/linux` or `/usr/src/linux-2.4` directory. Red Hat Enterprise Linux kernels are downloaded to a different directory, which is linked to `/usr/src/linux-2.4`. For more information on kernel sources, see Chapter 12.

Once you've identified the directory with your source code, there should be a `.config` file in that directory. If it isn't there, it means that this kernel has not been compiled for your computer. In that case, search this file for the `CONFIG_QUOTA` setting with the following command:

```
# grep CONFIG_QUOTA /usr/src/linux-2.4/.config
```

If the directory with your kernel source code is different, change this command accordingly. You should see one of the following results in the standard output:

```
CONFIG_QUOTA=y
CONFIG_QUOTA=n
```

In other words, quota support is active (`y`) or not active (`n`). If quota support is not active, you'll need to compile it into your kernel. See Chapter 12 for more information.

Note *The version number of the kernel should be associated with the settings you find in your bootloader configuration file, normally* `/boot/grub/grub.conf`*. For more information on the relationship between a bootloader and the kernel, see Chapter 12.*

User Quotas

To create a quota for specific users, follow these six basic steps:

1. Modify `/etc/fstab` to activate quota options for the filesystem of your choice.
2. Enable the change by remounting the filesystem.

3. Create the `aquota.user` file at the top of the subject filesystem. For example, if you're creating quotas on `/home`, create `/home/aquota.user`.
4. Scan the appropriate filesystem and create basic quota files with the `quotacheck` command.
5. Use `edquota` to apply quota limits for a specific user.
6. Finally, activate quotas with the `quotaon` command.

We explain these steps in more detail in the following sections.

Modifying /etc/fstab *and Remounting*

It's easy to modify `/etc/fstab` for quotas. Take a typical line from this configuration file, which in this case sets up `/home` as a filesystem on a separate partition:

```
LABEL=/home  /home  ext3  defaults  1 2
```

Fortunately, there's room in `/etc/fstab` to add the User Quota setting, `usrquota`. Space is scarce in `/etc/fstab`, since the boot process may not work if you let this code wrap to the next line. So, with the User Quota setting, this `/etc/fstab` line would read as follows:

```
LABEL=/home  /home  ext3  defaults,usrquota  1 2
```

Now you can activate the change by remounting the `/home` directory. Fortunately, you do not need to change runlevels or reboot with the rescue disk to make this work; all you need to activate `/etc/fstab` changes on `/home` is the following command:

```
# mount -o remount /home
```

Creating the Quota File

It's easy to create the quota file you need with the `touch` command. As we're creating quotas on the `/home` filesystem in this section, create an empty `aquota.user` file in the `/home` directory. The easiest way to do this is with the `touch` command:

```
# touch /home/aquota.user
```

It's important to set the security on this file so it's accessible only to the root user. Since this file need not be executable, you can do this with the following command:

```
# chmod 600 /aquota.user
```

Making the quotacheck

Now you're ready to create appropriate quota files with the `quotacheck -avum` command. This scans (`-a`) `/etc/mtab` for filesystems with enabled quotas, creates verbose (`-v`) output, looks for user quotas (`-u`), and remounts the scanned filesystem (`-m`).

Using edquota *for a User*

Next, you can set up quotas for a specific user. Run the `edquota` command for the user of your choice. For example, if you want to set quotas on user ez, run the following command:

```
# edquota ez
```

By default, this opens the quota information file for user ez in the `vi` editor, as shown here:

```
Disk quotas for user ez (uid 504)
  Filesystem blocks   soft   hard inodes soft hard
  /dev/hdd1   16852  18000  20000     26    0    0
```

As you can see, there are 16852 blocks of data (in KB) and 26 inodes used in ez's home directory. You can set hard and soft limits in each category. But what are hard and soft limits?

Soft limit A *soft limit* is the maximum amount of space or inodes allocated to a user. If there is no grace period, this acts as a hard limit. You can set a grace period with the `edquota -t` command.

Hard limit If there is a grace period, the *hard limit* is the absolute limit on the amount of space or inodes allocated to a user.

Now if you want to set a 100MB soft limit and a 110MB hard limit, edit the quota for ez to look like the following:

```
Disk quotas for user ez (uid 504)
  Filesystem blocks   soft   hard inodes soft hard
  /dev/hdd1   16852 100000 110000     26    0    0
```

Enabling Quotas

The last step, enabling quotas, is the simplest. You've already done the necessary configuration work. Just run the following command to enable quotas for all configured users on the `/home` filesystem:

```
# quotaon /home
```

Alternatively, you can deactivate quotas on the same filesystem with the `quotaoff /home` command.

GROUP QUOTAS

Creating group quotas is as easy as creating user quotas. The differences can be summarized in these same six steps:

1. Modify `/etc/fstab` to activate quota options for the filesystem of your choice. For group quotas, add the `grpquota` setting to the options for the target filesystem.
2. Enable the change by remounting the filesystem with the `mount -o remount` *`filesystem`* command.
3. Create the `aquota.group` file at the top of the subject filesystem. For example, if you're creating quotas on `/home`, create `/home/aquota.group`.

4. Scan the appropriate filesystem, and create basic quota files with the `quotacheck` command. Use the `-avgm` switches; `-g` configures group quotas.
5. Use `edquota` to apply quota limits for a specific group.
6. Finally, activate quotas with the `quotaon` command.

Activation in *rc.sysinit*

Once you've configured quotas in Red Hat Enterprise Linux, the operating system can take over the next time you reboot. Quota checking and activation commands are included in the default `/etc/rc.d/rc.sysinit` startup script. The relevant section is shown in Figure 9.12, which also attempts to convert the quota files associated with Linux kernel version 2.2 (`quota.user` and `quota.group`).

Figure 9.12
rc.sysinit activates quotas

```
# check remaining quotas other than root
if [ X"$_RUN_QUOTACHECK" = X1 -a -x /sbin/quotacheck ]; then
        if [ -x /sbin/convertquota ]; then
            # try to convert old quotas
            for mountpt in `awk '$4 ~ /quota/{print $2}' /etc/mtab` ; do
                if [ -f "$mountpt/quota.user" ]; then
                    action $"Converting old user quota files: " \
                    /sbin/convertquota -u $mountpt && \
                        rm -f $mountpt/quota.user
                fi
                if [ -f "$mountpt/quota.group" ]; then
                    action $"Converting old group quota files: " \
                    /sbin/convertquota -g $mountpt && \
                        rm -f $mountpt/quota.group
                fi
            done
        fi
        action $"Checking local filesystem quotas: " /sbin/quotacheck -aRnug
fi

if [ -x /sbin/quotaon ]; then
    action $"Enabling local filesysten quotas: " /sbin/quotaon -aug
fi
```

Applying Quotas to Other Users

You can set up common quotas for a number of different users. The `edquota` command allows you to set up the same quotas for a list of users. Assuming you've already set up quotas for user ez, the following command copies the identical limits for the other users that follow, in this case, mj, jm, and tp:

```
# edquota -up ez mj jm tp
```

Quota Monitoring

Now that you've set up quotas, you can get reports on who is using disk space and inodes and how much space they occupy. The `repquota` command gives you quota reports by users (`-u`) or groups (`-g`). You can also get a report on all filesystems with the `repquota -a` command.

If you want to check up on an individual user (`-u`) or group (`-g`), use the `quota` command. Individual users can check their own status with this command.

Creating User Private Groups

Red Hat Enterprise Linux has a unique way of organizing users and groups that promotes security. The following sections describe the Red Hat User Private Group scheme. You can use this scheme to configure a special secure group with a common directory.

The Red Hat Scheme

As noted in the beginning of the chapter, everyone's user ID number usually matches their group ID number in `/etc/passwd`. But this is generally true only for Red Hat Enterprise Linux and allied distributions. The other scheme is where every user has the same group ID number, which is usually 100. In other words, in other distributions every user belongs to the same group by default.

The Red Hat scheme is more suitable for a number of configurations. For example, it allows the users of an ISP to keep their files hidden from other users of that ISP. Yet you can still configure a shared directory for selected users.

Creating a Shared Directory

Sometimes you want users to be able to share files. Some users may be in a common department, or they may be working on a common project. You can set up a group and a directory where all imported files are readable by all members of that group.

The easiest way to illustrate this process is with an example. Say you need to set up a group and a shared directory for project members Tom, Adnan, Carlos, and Libby. In the following steps, you'll create the users, a common group, and a shared directory. Then you'll set the group ID (SGID) bit, which allows any user in the group to copy files to the shared directory and makes it readable by the other members of the group.

1. Give Tom, Adnan, Carlos, and Libby accounts on your system with the `useradd` *`username`* command. Remember to assign passwords to each user.
2. Use the `groupadd project` command to create the project group. Edit `/etc/group` to add your new users to that group.
3. Set up a new shared directory, called `/home/project`. Give it full permissions (`rwx`) for the user and group that own this directory with the `chmod 770 /home/project` command.
4. Configure the SGID bit on the directory with the `chmod g+s /home/project` command. This allows all users in the group that owns the directory to have ownership-level permissions.
5. Set up appropriate ownership for that directory with the `chgrp project /home/project` command.
6. Feel free to log in as one of the users. Copy files from the home directory of a user to `/home/project`. Log in as a different user in the same group. Can you do anything with the file copied by the first user?

Tip *It's possible to combine the two* `chmod` *commands; the* `chmod 2770 /home/project` *command configures the noted permissions and adds the SGID bit to that directory.*

Summary

In this chapter, we examined the basics of how users and groups are managed in Red Hat Enterprise Linux. We began with the configuration files. While `/etc/passwd` and `/etc/group` contain basic information about users and groups, `/etc/shadow` and `/etc/gshadow` include encrypted passwords and password age parameters in more secure files. New users are assigned a home directory with a copy of the files in `/etc/skel`, based on the parameters shown in `/etc/login.defs`.

You can create users and groups directly, by editing the appropriate configuration files. You can create them more efficiently with commands such as `useradd` and `groupadd`. Users and groups can be deleted with the nearly parallel `userdel` and `groupdel` commands. And you can manage how user passwords are regulated with the `chage` command. Alternatively, you can configure users and groups with the Red Hat User Manager, which you can start with the `redhat-config-users` command.

As an administrator in the enterprise, you need not take full responsibility for your computers or network. You can configure partial administrative privileges for users in `/etc/sudoers`. You can also limit access to the root account by adding privileged users to the wheel group in `/etc/group`, and activating the appropriate command in `/etc/pam.d/su`.

This system of users, groups, and associated commands is known as the Shadow Password Suite. With the appropriate strong passwords, this suite can improve the security of your user and group accounts. The `pwconv` and `grpconv` commands convert `/etc/passwd` and `/etc/group` to conform to this suite. The `pwunconv` and `grpunconv` commands reverse this process.

You can manage the demands of your users with quotas. Linux quotas can limit users by inodes or by the space their files occupy on a specific partition. Quotas are easily configurable once you've modified `/etc/fstab` to incorporate quotas on desired filesystems. And once they're configured, Red Hat Enterprise Linux automatically activates your quotas when it boots.

Finally, the Red Hat User Private Group scheme provides additional security by isolating every user in his or her own unique individual group. However, you can still organize users in a common group with a shared directory.

In the next chapter, you'll learn all about the Red Hat way of managing packages with the Red Hat Package Manager. This system has been so successful that it has been adapted by a number of other competitive Linux distributions.

Chapter 10

Managing and Updating Packages with RPM

The Red Hat Package Manager (RPM) provides a standardized way to group the software you need for various utilities and applications. RPMs make it possible for Red Hat to organize its Enterprise Linux distribution into more than 1,000 packages instead of tens of thousands of files.

You'll find that using RPMs to add new programs and applications is an easy process. The RPM is so successful that it has been adapted as the primary package manager by other competitive Linux distributions, including SUSE and Mandrake.

As an administrator, you'll need to install, upgrade, remove, and maintain many different RPM packages. RPMs also include dependency information, which helps you install any prerequisite packages you may need. When Red Hat adds new features or provides more secure software, you may want to upgrade what you have as well.

With the way Red Hat organizes package groups, the Red Hat GUI Package Management tool may sometimes be a more efficient way to manage your system. We'll show you how you can configure it to point to the network source you used to install Red Hat Enterprise Linux.

Although the RPMs that you install are in binary format, Red Hat provides the source code for each package. You can use the `rpmbuild` command to organize and build these packages into the binary files that anyone can install. Alternatively, you can build binary RPMs from the other standard package system, organized as the *tarball*.

One of the advantages of RPMs is that you can verify the integrity of packages and files. If a file has been modified without your knowledge, the correct `rpm` command identifies the altered file.

The RPM system is rich with features. This chapter just scratches the surface, providing what I think are the most important RPM skills to the Linux administrator.

Red Hat also stores the latest Enterprise RPMs on the Red Hat Network. Alternatively, you can download the development packages available through the Fedora Project. You can use `up2date` through the Red Hat Network to update the RPMs of your choice based on a current database of upgradeable RPMs. This chapter covers the following topics:

- Installing and upgrading, simplified
- Using the Red Hat GUI Package Management Tool

- Making source RPMs work
- RPM security
- Updating RPMs

Installing and Upgrading, Simplified

Installations and upgrades form the essence of managing RPM packages. When you install an RPM, you're adding new software to your system. When you're upgrading an RPM, you're updating the associated software with the latest features.

Before you install or upgrade an RPM, you should know whether the desired package is already on your system. The RPM query can also give you descriptive information about the package, and it can verify and list the files associated with the package.

You can install or upgrade RPMs from local or remote sources. There are provisions to include username/password combinations when you access a binary RPM from a remote location.

RPM packages include dependency information. For example, the kernel source code RPM needs the GNU C language compiler. Since the kernel source is dependent, you should install the compiler before installing the source code.

If you're looking for a specific file, install the standard Red Hat Enterprise Linux database of RPMs. This can help you identify packages that you might need to install.

Queries

The query mode of the `rpm` command has many dimensions. In its simplest form, you can run this command to find the version of an installed package. With additional switches, you can use it to view summary information, list files, verify contents, and more.

A Simple Query

The simplest query takes the form of `rpm -q` *`packagename`*. For example, you can locate the installed version of the setup RPM, which contains a number of basic configuration files. Just run the following command:

```
# rpm -q setup
setup-2.5.27-1
```

Information Queries

Queries can provide more information about a package. For example, the `rpm -qi` *`packagename`* command helps you get the summary information associated with the `setup` RPM. The output, as shown in Figure 10.1, can reveal a lot of good information about a specific package.

Identifying the Owner

Suppose you've heard that upgrades are available for a certain file, but you don't know what RPM package to use. In this case, just use the `rpm -qf` *`filename`* command to identify the name of the

package. For example, if you need to identify the RPM package that owns the `/etc/passwd` configuration file, run the following command:

```
# rpm -qf /etc/passwd
setup-2.5.27-1
```

Note that you need the full path to the file in question.

Listing the Files in an RPM

If you're not sure about a package, you can list the files within by using the `rpm -ql packagename` command. That list confirms whether certain configuration files or commands are part of that package. If you're upgrading, that information can help you understand what is at risk when you upgrade. Listing the files in the `setup` RPM provides a good example, as shown in Figure 10.2.

Figure 10.1
An RPM package summary information

```
[root@Enterprise3 root]# rpm -qi rpm
Name        : rpm                          Relocations: (not relocateable)
Version     : 4.2.1                             Vendor: Red Hat, Inc.
Release     : 4.2                           Build Date: Thu 25 Sep 2003 03:26:56
 PM EDT
Install Date: Thu 23 Oct 2003 03:21:18 PM EDT      Build Host: bugs.devel.redhat
.com
Group       : System Environment/Base       Source RPM: rpm-4.2.1-4.2.src.rpm
Size        : 4769124                          License: GPL
Signature   : DSA/SHA1, Thu 25 Sep 2003 05:14:10 PM EDT, Key ID 219180cddb42a60e
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary     : The RPM package management system.
Description :
The RPM Package Manager (RPM) is a powerful command line driven
package management system capable of installing, uninstalling,
verifying, querying, and updating software packages. Each software
package consists of an archive of files along with information about
the package like its version, a description, etc.
[root@Enterprise3 root]#
```

Figure 10.2
A list of files in an RPM package

```
[root@Enterprise3 root]# rpm -ql setup
/etc/bashrc
/etc/csh.cshrc
/etc/csh.login
/etc/exports
/etc/filesystems
/etc/group
/etc/host.conf
/etc/hosts.allow
/etc/hosts.deny
/etc/inputrc
/etc/motd
/etc/passwd
/etc/printcap
/etc/profile
/etc/profile.d
/etc/protocols
/etc/securetty
/etc/services
/etc/shells
/usr/share/doc/setup-2.5.27
/usr/share/doc/setup-2.5.27/uidgid
/var/log/lastlog
[root@Enterprise3 root]#
```

TIP Package upgrades are always a risk. If you've configured a daemon and then overwrite it with an upgrade, you may lose your custom configuration. While RPM is supposed to save customized configuration files with the `.rpmsave` *extension, it is still a good practice to back up key configuration files. In some cases, installing two versions of the same package is safer than upgrading. You'll see an example of where this is true when you upgrade the Linux kernel in Chapter 12.*

RPMs and CPUs

Many RPMs are built for specific CPUs. For example, many RPMs have an extension such as `.i386.rpm` or `.noarch.rpm`. While Red Hat Enterprise Linux 3 can't be installed on computers with Intel 386–level CPUs, RPMs with the `.i386.rpm` extension can be installed on all Red Hat Enterprise Linux computers with Intel-compatible Pentium-level CPUs.

RPMs with other extensions are optimized for their CPUs. When possible, you should install the RPM associated with your CPU. To find your CPU identifier, type the `uname -p` command. Some of the popular extensions are described in Table 10.1. `noarch.rpm`, for a CPU-independent installation, `i586.rpm`, `i686.rpm`, `ia64.rpm`, and `sparc.rpm`.

TABLE 10.1: RPM EXTENSIONS

Extension	CPU
`noarch.rpm`	Doesn't depend on the CPU; generally can be installed on all computers
`i586.rpm`	For computers with many Intel 32-bit Pentium systems
`i686.rpm`	For computers with many Intel 32-bit Pentium systems; often applies to users with AMD 32-bit CPUs
`ia64.rpm`	For computers with many Intel 64-bit Itanium systems
`sparc.rpm`	For computers with the Sun Microsystems SPARC CPU

The Basic Installation

Installing a new RPM package is simple. Just use the `rpm -i` *`packagename-versionnumber`* command, and if that package is not already on your system, it is automatically installed. In the enterprise, you may be installing new RPMs from a network source based on the Red Hat Enterprise Linux installation CDs. For example, if you have an NFS connection such as that used in Chapter 4 to install over a network, you can still connect to that source after installation with a command such as the following:

```
# mount -t nfs server.example.com:/mnt/inst /mnt/inst
```

Make sure your NFS server is active, and there's no firewall blocking communication. Substitute the name or IP address of your network installation server for `server.example.com`. Once the source is mounted, you may run the following command to install the `setup` RPM package:

```
# rpm -i /mnt/inst/RedHat/RPMS/setup-*
```

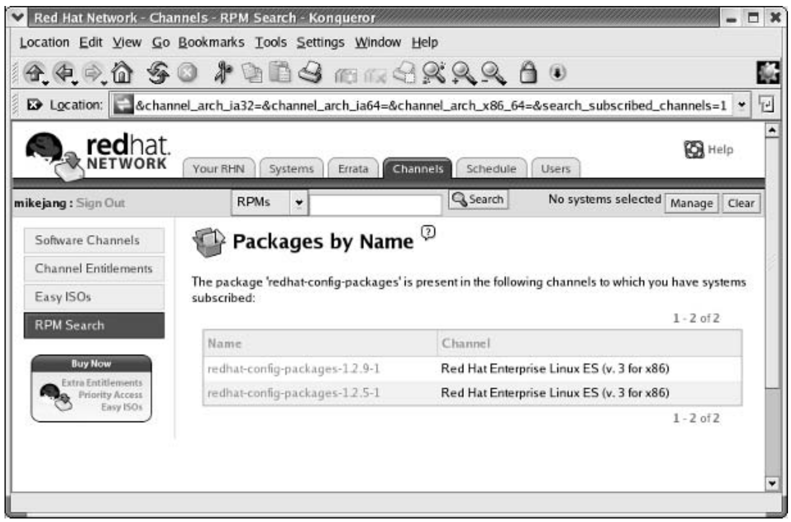
You may even be installing them directly from the CDs, mounted on `/mnt/cdrom`; in this case, substitute `/mnt/cdrom` for `/mnt/inst`. The asterisk is an appropriate wildcard because RPM packages are

updated frequently, but the actual name of the package usually remains constant. If multiple packages start with `setup-*`, this command installs all of them.

Red Hat Network Downloads

If you have an official subscription to Red Hat Enterprise Linux, you should also have access to the Red Hat Network downloads for the packages of your choice. For example, when you log into `rhn.redhat.com`, you can search for the RPMs of your choice. The Red Hat Network lists the packages you're allowed to download. For example, Figure 10.3 lists the two versions of the `redhat-config-packages` RPMs that I could download on my computer.

Figure 10.3
Downloading RPMs from the Red Hat Network



The latest version of this and many other upgraded RPM packages are also available using the quarterly Updates CD, described in Chapters 3 and 4, or using the `up2date` utility, described later in this chapter. However, you don't even need a Red Hat Network subscription to download this update.

Downloads from Alternate Sources

The groups that have created the Red Hat Enterprise Linux 3 rebuilds have published their RPMs, built from the Red Hat source code, on their servers. For example, the cAos project have made their rebuilt RPMs freely available. You can download the rebuilt version of the updated `redhat-config-packages` RPM from the cAos project servers at `www.caosity.org`.

Alternatively, if you want to download the development version of this same package, you can do so from the Development directories available through the Fedora Project. The Fedora version of this RPM is known as `system-config-packages`.

In either case, you can install the RPMs of your choice directly from the source. For example, if you want to install the `elinks` RPM package from the `RedHat/RPMS` directory on a server named `ftp.example.com`, run the following command:

```
# rpm -ivh ftp://ftp.example.com/RedHat/RPMS/elinks-*
```

As the `-i` extension installs, the `-v` and `-h` extensions set up verbose output with hashing, so you can monitor the progress of the installation. Some FTP servers require usernames and passwords. If you were installing the `elinks` RPM package, with a username of anonymous and a password of efgh, you could use the following command:

```
# rpm -ivh ftp://anonymous@ftp.example.com/pub/fedora/linux/core
➥/development/i386/Fedora/RPMS/elinks-*

Password for anonymous@ftp.example.com:
```

The password you enter at the prompt is not shown on the screen. While you could add the password to the `rpm` command, it isn't advisable, since the password would appear on your screen and be transmitted in clear text over the Internet.

You can even use this command to install the latest version of multiple packages. But this command often does not work over the Internet; if you want to install a package reliably, download it first. The Development packages discussed near the end of this chapter are the latest packages available from Red Hat.

Upgrades

There are always risks associated with upgrades. You may accidentally overwrite configuration files you've customized for your computer and/or your network. Or perhaps the upgraded software has interaction problems with other applications installed on your system.

However, there are often good reasons to upgrade an RPM package. Sometimes, you or your users need updated features. You may be upgrading software to address a security alert. Or you may need upgraded software (such as compilers) to handle upgraded versions of other new packages, such as kernels.

Two switches are associated with upgrades: `-U` and `-F`. Both switches can upgrade an RPM package. The difference is what happens if there is no installed RPM package to upgrade. In that case, the `rpm -U` *`packagename`* command installs the new package, and the `rpm -F` *`packagename`* command does not.

Generally, it is a good practice to include the `-v` (verbose) and the `-h` (hash mark) switches whenever you upgrade or freshen an RPM package. For example, if you're upgrading an installed version of the `redhat-config-packages` RPM from a mounted Updates CD, the following command can help you monitor the progress of the installation (with hash marks), as well as any error messages that may appear:

```
# rpm -Uvh /mnt/cdrom/RedHat/Updates/redhat-config-packages-*
```

Dependencies

When you try to install or upgrade an RPM, you may get an error message. Perhaps the most common rpm error message is based on dependencies.

An RPM dependency occurs when one package will not work unless a different package is already installed. The source code for the package lists other RPM packages that it needs—in other words, packages that it depends upon. You can see an example of a dependency from when I tried to install the kernel-source RPM:

```
# rpm -Uvh /mnt/inst/RedHat/Updates/kernel-source-*
warning: /mnt/cdrom/RedHat/RPMS/ kernel-source-2.4.21-9.EL.i386.rpm: Header V3 DSA
error: Failed dependencies:
   gcc >= 2.96-98 is needed by kernel-source-2.4.21-9.EL
Suggested resolutions:
   gcc-3.2.3-20.i386.rpm
```

The output suggests that I need to install the gcc (GNU C Compiler) package first. You could install both packages simultaneously or install gcc first. If this seems like a lot of trouble, you could also use the rpm --nodeps switch to ignore the dependency. As long as you install gcc before you actually use the kernel-source package, this should not be a problem. One way to do that is with the following commands:

```
# rpm -Uvh --nodeps /mnt/inst/RedHat/Updates/kernel-source-*
# rpm -Uvh /mnt/inst/RedHat/Updates/gcc-3*
```

However, this solution is not perfect; various gcc-3* RPMs also depend on other packages. You can satisfy these dependencies one at a time, or you can do it all at once with the Red Hat GUI Package Management tool that we describe later in this chapter. While I prefer the command-line interface for most operations, you'll see how the GUI tool is more efficient than the following set of commands:

```
# cd /mnt/inst/RedHat/Updates/
# rpm -Uvh gcc-* cpp-* libgcc-* libf2c-* libgnat-* libobjc-*
```

Deletions

It's easy to delete an RPM package by using the -e switch. You don't even have to know the version number of the package. For example, the following command deletes the kernel-source RPM:

```
# rpm -e kernel-source
```

Since you do not need to cite the path to delete an RPM, it is easy to delete multiple packages with the same command:

```
# rpm -e kernel-source gcc
```

A Database of RPMs

Say you're looking for a file or a command and discover that it isn't yet installed on your computer. You know that Red Hat Enterprise Linux files are organized by RPM packages. In some cases, it isn't too difficult to figure out the right RPM to install; for example, commands such as `smbclient` are part of the `samba-client-*` RPM package. However, if you're looking for the RPM associated with some obscure program library, finding the right package can be more difficult.

This is where the Red Hat Enterprise Linux database can help. Once installed, the `rpmdb-redhat-*` RPM can help you find the RPM package associated with every file that you can install in the current version of Red Hat Enterprise Linux.

As an example, if you're looking for the package associated with `/etc/exports`, the following command will work, once the `rpmdb-redhat-*` RPM package is installed:

```
# rpm --redhatprovides /etc/exports
setup-versionnumber
```

Extracting a Single File

Sometimes all you want to do is extract a single file from an RPM package. With the `rpm2cpio` and `cpio` commands, this is a simple process. For example, assume you've accidentally deleted the main Samba configuration file, `/etc/samba/smb.conf`. As you can see from the following command, it's part of the `samba-common` RPM:

```
# rpm --redhatprovides /etc/samba/smb.conf
samba-common-3.0.0-14.3E
```

You can extract this file from the `samba-common` RPM. First, assume that the network installation source is mounted on the `/mnt/inst` directory. You can inspect a list of files in the `samba-common` RPM with the following command:

```
# rpm2cpio /mnt/inst/RedHat/RPMS/samba-common-* | cpio -it
```

For the Samba configuration file in question, I've added one more qualifier to limit the list to the desired files:

```
# rpm2cpio samba-common-* | cpio -it | grep conf
```

For this particular RPM, I see the `smb.conf` file in the `./etc/samba` directory. I therefore use the following command to extract the desired `smb.conf` file in the `etc/samba/smb.conf` subdirectory:

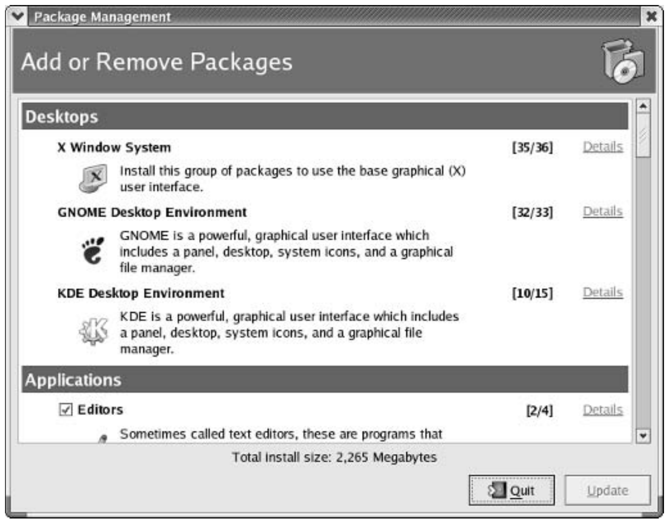
```
# rpm2cpio amba-common-* | cpio -imd ./etc/samba/smb.conf
```

Note, there is no starting slash; in other words, `etc/samba/smb.conf` is relative to the current directory. If you've run this command as the root user in the `/root` directory, you should now find `smb.conf` in the `/root/etc/samba` directory. Now you can copy this file to its original location in the `/etc/samba` directory and continue configuring Samba on your computer.

Using the Red Hat GUI Package Management Tool

You can use the Red Hat GUI Package Management tool, also known as `redhat-config-packages`, to inspect, install, and remove the RPM packages currently on your Linux system. Start it from a GNOME desktop by selecting Main Menu ➤ System Settings ➤ Add/Remove Applications. This opens the Package Management window, shown in Figure 10.4.

FIGURE 10.4
The Red Hat GUI Package Management tool



If you installed Red Hat Enterprise Linux graphically per Chapter 3, this tool should look familiar. It includes the same organization of package groups you used during the graphical installation process. By default, it's configured to look for RPMs on Red Hat Enterprise Linux CDs. But that's not very useful in the enterprise, which is why we'll show you how to configure access to a network installation source.

Configuring Access to a Network Installation Source

Assuming you've added the `.discinfo` file to the network installation source as described in Chapters 3 and 4, you can point the Package Management tool to that source. It's a straightforward process. First, mount the network installation source on a local directory; second, use the `redhat-config-packages --tree` command to point to that directory. Assuming you've shared the installation source via the shared NFS directory described earlier, you can mount the source with the following command:

```
# mount -t nfs server.example.com:/mnt/inst /mnt/inst
```

You don't need to repeat this command if you've already run it during this session. It works for a Samba source equally well.

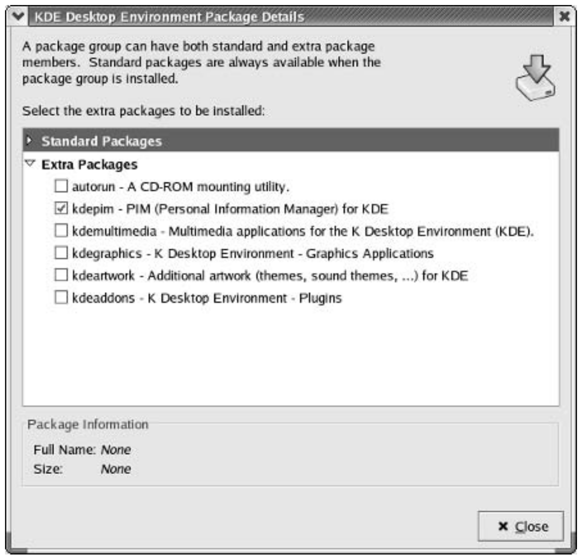
Open a command-line interface inside the GUI. You can then point the Package Management tool to this source with the following command:

```
# redhat-config-packages --tree=/mnt/inst
```

Managing Packages by Group

You can select *some* individual packages in each group for installation and removal. As an example, take a look at the packages associated with the KDE Desktop Environment. On the far right side of the associated entry, click Details. This opens the KDE Desktop Environment Package Details window, shown in Figure 10.5.

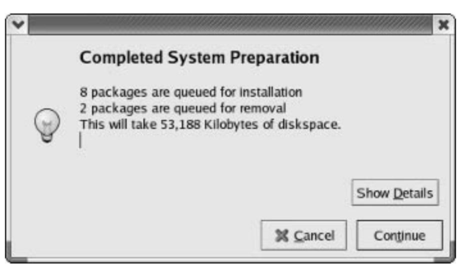
FIGURE 10.5
KDE Desktop Environment Package Details window



As you can see, there are two categories of packages: standard and extra. Standard packages correspond to the mandatory packages as defined in the `comps.xml` file. The extra packages are either default or optional packages as defined in `comps.xml`.

In this way, you can select or deselect the packages or package groups of your choice. Make any desired changes, and click Close. When you click Update in the Package Management window, this utility makes sure you don't have unsatisfied dependencies. You get a last chance to cancel (see Figure 10.6) before the changes are made. Click Show Details to review the changes to be made.

FIGURE 10.6
Change summary



Making Source RPMs Work

A key feature of Linux is the easy accessibility to the source code. Since Red Hat Enterprise Linux is built on RPM packages, that means access to the *source RPMs* (SRPMs). An SRPM includes the code and instructions needed to build a *binary RPM*, which you can then install on your Red Hat Enterprise Linux computer.

You now need to use the rpmbuild command to process RPM source code.

To understand how to use a source RPM, you need to know the default directory structure and understand that `.spec` files are used to build a binary RPM.

Directories

By convention, SRPMs are easy to identify; they have the `.src.rpm` extension. SRPMs include specification and other files, which you can set up in various `/usr/src/redhat` subdirectories. Any SRPM you build into a binary file is also set up in the same directory structure. The five key SRPM directories are shown in Table 10.2.

TABLE 10.2: SOURCE RPM DIRECTORIES

DIRECTORY	FUNCTION
`/usr/src/redhat/BUILD`	Any source code that you process is unpacked and built here.
`/usr/src/redhat/RPMS`	Binary RPMs that you create from an SPRM are found here.
`/usr/src/redhat/SOURCES`	Contains the actual source code.
`/usr/src/redhat/SPECS`	Includes the files that control the RPM build process.
`/usr/src/redhat/SRPMS`	Includes SRPMs created during the build process.

To break an SPRM down into these directories, you need to install it. For example, if you want to work with the `anaconda-product` package, you'll need to install the associated `.spec` file. For example, if you've mounted an SRPM Red Hat CD, you can do so with the following command:

```
# rpm -i /mnt/cdrom/RedHat/SRPMS/anaconda-product*.src.rpm
```

The Spec File

The key to managing an SRPM is in its spec file. Once you've installed an SRPM, you should be able to find its spec file in the `/usr/src/redhat/SPECS` directory. This file controls how packages are built and configures commands when an RPM is installed or deleted.

The key sections in a spec file are `%prep`, `%build`, and `%install`. They allow you to build source and binary RPMs. One important variable is *`Requires`* or *`BuildRequires`*, which lists other packages you should install first. Other typical sections in a `.spec` file include the following:

%define Includes basic parameters such as the location of the top-level directory for that package. For example, you may see a `ROOT /var/ftp` line in this section. The section includes basic summary

information for the RPM. When this RPM is installed, this is what a user will see in response to the `rpm -qi` *`packagename`* command.

%package Lists packages that depend on this particular RPM.

%description Provides more information for the `rpm -qi packagename` command.

%prep Includes preparation commands for archives and patches.

%setup Contains processing commands for unpacking archives.

%build Builds the code to be compiled.

%install Adds the commands that actually build the files and install the package in well-defined directories.

%clean Includes basic commands for deleting any intermediate files from your system.

%post Contains postinstallation scripts, such as a script that modifies a user account.

%postun Contains scripts after you remove a package.

%pre Contains preinstallation scripts, such as a script that prepares a directory.

%preun Contains scripts before you remove a package.

%triggerin Contains parts of other packages you've copied.

%config Lists configuration files for `/etc`.

Spec files are not as difficult as they may look. For the most part, they include regular Linux commands and descriptions, which you can modify in a text editor.

Building Binaries from a Tarball

You can create an RPM from a tarball. But first, you need a spec file. As you've seen in the previous section, this can be a little difficult.

Note *A* tarball *is a single file that's a package, or an archive, of a group of files. When you "unpack" a tarball, the files in the package are copied to the computer. In that way, a tarball is similar to a Microsoft Windows compressed "zip" file archive. Tarballs are typically available in a compressed format, with extensions such as* `.tar.gz`*,* `.tgz`*, and* `.tar.bz2`*.*

One way to learn more about this process is to read different spec files. For example, take the following excerpts from a `dosemu.spec` file:

```
%define vimver 5.8
%define vim vim58
Summary: A DOS emulator.
Name: dosemu
Version: 1.1.1
Release: 3
Exclusivearch: %{ix86}
```

```
License: distributable
Group: Applications/Emulators
Source0: ftp://ftp.dosemu.org/dosemu/dosemu-%{version}.tar.bz2
…
Patch0: dosemu-0.66.7-config.patch
…
%package -n xdosemu
Requires: dosemu = %{PACKAGE_VERSION}
Summary: A DOS emulator for the X Window System.
Group: Applications/Emulators
```

This file was taken from a `dosemu-*.src.rpm` package. The system is fairly straightforward; from the code, you can find the release version, the URL for the source and related patch(es), and the summary description. You can also see that the `xdosemu` package requires `dosemu`, which sets up an RPM dependency.

Building a Binary RPM

All you need to create a binary RPM is the source code (which you can get from a source RPM or a tarball) and a spec file. You can create a spec file from scratch or modify an existing spec file from a source RPM.

There are two basic ways to build a binary RPM:

```
# rpmbuild -ba packagename.spec
# rpmbuild -bb packagename.spec
```

The first command (`rpmbuild -ba`) creates binary and source RPM packages. The second command (`rpmbuild -bb`) creates only the binary RPM package.

NOTE *The* `rpm -ba` *and* `rpm -bb` *commands no longer work starting with Red Hat Enterprise Linux 3. Their functionality has been moved from the* `rpm-build-*` *RPM to the* `rpmbuild` *command.*

RPM Security

Once you've learned to use RPMs, it's easy to just install and forget them and not worry about security. A cracker may post a virus or a Trojan horse on an RPM posted online. The `rpm` command includes ways to check the integrity of an RPM, using the Pretty Good Privacy (PGP) system (see the next section). You can also verify the contents of a package, or even a specific file.

RPM and Pretty Good Privacy

The RPM system uses one of the security standards associated with e-mail security, known as Pretty Good Privacy (PGP). Developed by Phil Zimmerman, PGP provides a private-key and public-key system. With Red Hat Enterprise Linux, the GNU way of using PGP is known as the GNU Privacy Guard (GPG).

The key to all of this is the Red Hat GPG key. It should be installed by default as `/usr/share/doc/rpm-`*`version`*`/RPM-GPG-KEY`. If it isn't there, you can get it from at least one of the following sources:

- From the Red Hat Enterprise Linux installation CDs, in the main directory. If you install CDs in the default location, the key will be in `/mnt/cdrom/RPM-GPG-KEY`.
- From `www.redhat.com`. As of this writing, different keys are available at `www.redhat.com/solutions/security/news/publickey.html`.

Next, import the GPG public key. For example, if you're importing from the installation CD, you should import to the `/var/lib/rpm/Pubkeys` file with the following command:

```
# rpm --import /mnt/cdrom/RPM-GPG-KEY
```

NOTE *The* `rpm --import` *command is fairly new. If you're using an older version of Red Hat Linux (before 7.3), you may need to use the* `gpg --import publickey` *command.*

Verifying a Package

Now you can verify any RPM package for a genuine Red Hat Enterprise Linux signature. For example, it may be a good idea to verify the integrity of the kernel sources before recompiling. To perform this task on a `kernel-sources` RPM in the `/mnt/inst/RedHat/Updates` directory, run the following command:

```
# rpm -K /mnt/inst/RedHat/Updates/kernel-source-*.rpm
/mnt/inst/RedHat/Updateskernel-source-versionnumber.rpm: (sha1) dsa sha1 md5 gpg OK
```

This verifies the integrity of the `kernel-source` RPM to the noted encryption schemes, including GPG.

Verifying a File

It's useful to check files against the original configuration. For example, if a cracker changes a file on your computer, you want to know about it. There are a number of standard things about every file you can check against the original. The data associated with every file installed through an RPM package is stored in the RPM database in the `/var/lib/rpm` directory.

If you suspect that a certain command isn't working quite as it should, you can check it against the RPM database. Take the `mount` command as an example. You can check the integrity of `mount` with the following command:

```
# rpm -Vf /bin/mount
```

If you don't see any output, the command matches what was originally installed.

NOTE *In Red Hat Enterprise Linux 3, the* `rpm -Vf` **`/path/to/file`** *command also checks the integrity of the other files associated with the associated RPM package.*

Alternatively, if someone tampered with the `mount` command, you may see output similar to the following:

```
# rpm -Vf /bin/mount
SM5....T    /bin/mount
```

This command checks nine attributes of `/bin/mount`. If you see one of the letters shown in Table 10.3, the file differs from the original in some way. In this particular case, there are changes to the file size, permissions, the MD5 checksum, and the file modification time.

TABLE 10.3: ***RPM*** **FILE VERIFICATION ISSUES**

OUTPUT	FAILED TEST
S	File size mismatch
M	Mode (different permissions and file type)
5	MD5 checksum wrong
L	Symbolic link incorrect
D	Device number wrong
U	User ownership changed
G	Group ownership changed
T	File modification time mismatch
?	Unreadable file
c	Configuration file flag

In some cases, a "failure" is not a problem. For example, if you've revised your `/etc/inittab` file, you'll see what looks like a verification failure:

```
# rpm -Vf /etc/inittab
..5....T c /etc/inittab
```

However, this particular "failure" may not mean that a problem exists. For example, I got this result after modifying the `initdefault` variable in this configuration file. In other words, the checksum (`5`) changed because I changed the content of the file; and the file modification time (`T`) is different from when Red Hat Enterprise Linux was installed on my computer.

When I ran the previous command on my Red Hat Enterprise Linux computer, I got the following surprise:

```
# rpm -Vf /etc/inittab
S.5....T c /etc/X11/prefdm
..5....T c /etc/inittab
```

I didn't request information about the `/etc/X11/prefdm` file, yet I'm told it's been changed. Actually, this is as designed for Red Hat Enterprise Linux. As this file is part of the same `initscripts` RPM

package, I realize that this command checks the integrity of *all* the files in this package. Because the `initscripts` RPM is installed by default, you can get a list of files associated with that RPM using the following command:

```
# rpm --ql initscripts
```

Updating RPMs

There are several ways you can update RPMs on your Red Hat Enterprise Linux computer. We've described some of them earlier in this chapter. The following is a general list of databases of RPMs designed for Red Hat operating systems:

- Red Hat Network for official subscriptions
- Fedora for "bleeding-edge" Red Hat RPMs
- "Rebuild" servers such as cAosity, White Box Linux, and Tao Linux
- Servers with older versions of Red Hat Linux

If you have an official subscription to Red Hat Enterprise Linux, use your Red Hat Network account. You'll be able to download the latest RPMs optimized for the enterprise and supported by Red Hat.

If you want the latest RPMs designed for Red Hat operating systems, use the packages developed for Fedora Linux. Red Hat has stated that it's using Fedora as a testing ground for future Red Hat Enterprise Linux distributions.

If you don't have an official subscription to Red Hat Enterprise Linux, you can use a rebuild version of this operating system. As of this writing, Red Hat has been updating its servers with the latest available Red Hat Enterprise Linux 3 RPMs, rebuilt from the freely available source code.

Red Hat Enterprise Linux (and some of the "rebuilds") allows you to update your subscribed computer systems automatically using `up2date`. If you have another system, you can upgrade your software from the source RPMs, one package at a time. You can also upgrade your software from one of the other RPM databases, one package at a time. Alternatively, you can use the Yellow dog Updater, Modified, also known as `yum`.

If you want to upgrade a specific RPM, download it to a directory such as `/tmp`. Back up your current configuration as it relates to that package. If possible, use the `rpm` command to install (`-i`) and not upgrade (`-U`) the new package. If you have a problem, it's easier to restore the original configuration. You'll see an example of this process when you learn to upgrade the Linux kernel in Chapter 12.

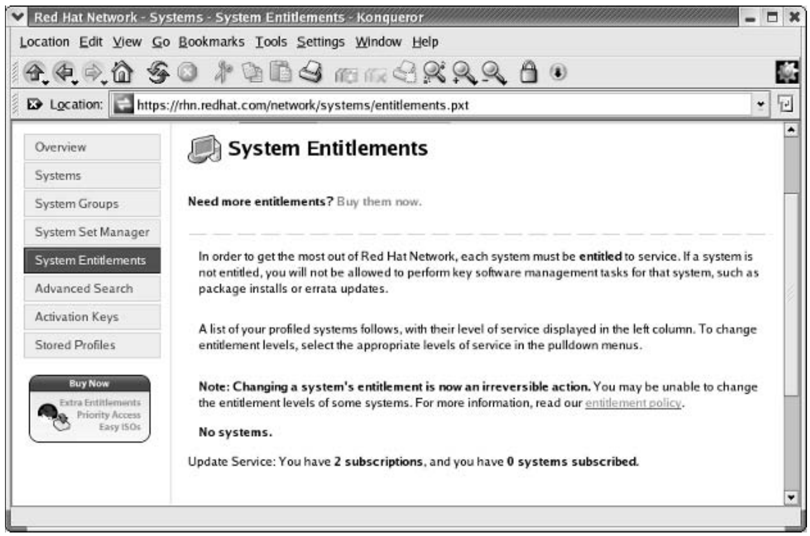
WARNING *If you have a subscription to Red Hat Enterprise Linux, you're allowed to install updates only on authorized computers. Although I'm not a lawyer, my understanding of the Red Hat Enterprise support contract suggests that Red Hat can cancel your subscriptions if you install downloaded RPMs from the Red Hat Network on unauthorized computers.*

The Red Hat Network

When you get an official copy of Red Hat Enterprise Linux, it comes with support and a subscription to the Red Hat Network. Red Hat probably e-mailed you a username and password that you can use to log into the Red Hat Network. It may be set up on the same account you used for an older distribution, such as Red Hat Linux 9.

Navigate to `rhn.redhat.com`. Make sure you can log into the network. Once you're logged in, click the Systems tab, and then click System Entitlements in the left pane, as shown in Figure 10.7.

FIGURE 10.7
The System Entitlements option



If your username and password does not work, or you don't see the subscriptions you bought, check your subscription e-mail or contact Red Hat support. As of this writing, the contact points are `customerservice@redhat.com` or 1-866-273-3428. Additional contact information is available from `www.redhat.com/about/contact/directory.html`.

CONFIGURING *UP2DATE*

Before you proceed with updating your system, you should understand the defaults associated with `up2date`. In either the GUI or the text console, you can configure `up2date` with the `up2date-config` command. It's easier to illustrate the result in the GUI; this command opens the Red Hat Network Configuration window shown in Figure 10.8. As you can see under the General tab, you can configure a network server of your choice and any proxy servers configured on your local network.

When you click the Retrieval/Installation tab shown in Figure 10.9, you'll be able to configure how packages are retrieved, installed, and stored. We've detailed the options in Table 10.4.

FIGURE 10.8
Configuring a network server

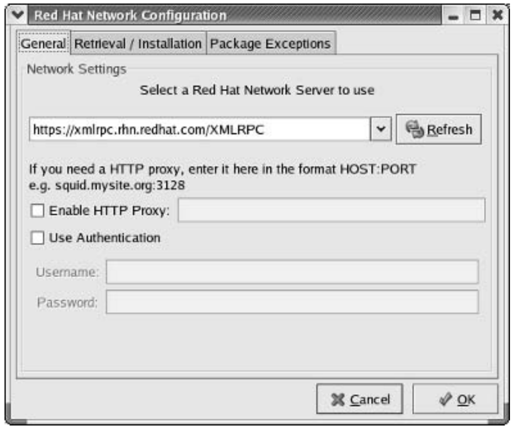


FIGURE 10.9
Specifying package update characteristics

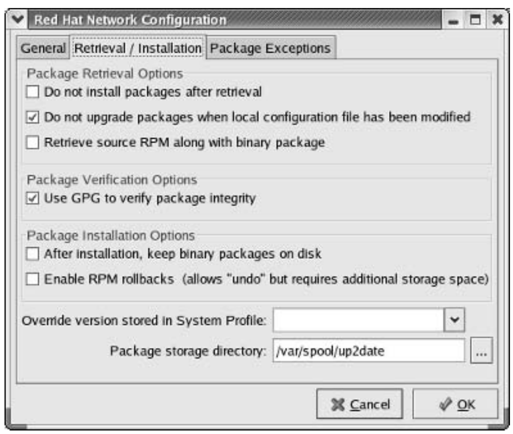


TABLE 10.4: RED HAT NETWORK RETRIEVAL/INSTALLATION OPTIONS

OPTION	DESCRIPTION
Do Not Install Packages After Retrieval	Downloads newer packages from the database to the specified package storage directory; doesn't install those packages.
Do Not Upgrade Packages When Local Configuration File Has Been Modified	Doesn't upgrade the package of a service you've configured. This allows you to test the upgraded package in a controlled manner.
Retrieve Source RPM Along With Binary Package	Downloads the source code along with each package.

Continued on next page

TABLE 10.4: RED HAT NETWORK RETRIEVAL/INSTALLATION OPTIONS *(continued)*

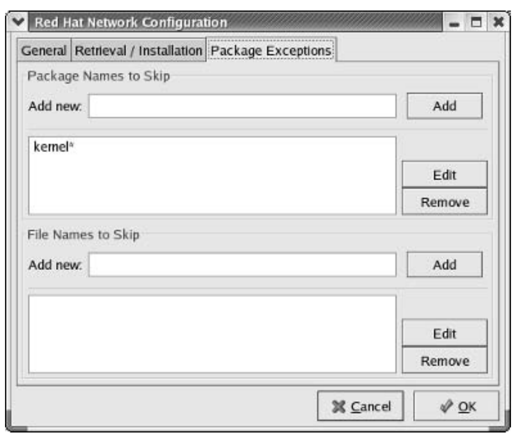
OPTION	**DESCRIPTION**
Use GPG To Verify Package Integrity	Checks each downloaded package using GNU Privacy Guard (see Chapter 17).
After Installation, Keep Binary Packages On Disk	Keeps the RPM package after installation.
Enable RPM Rollbacks	Allows you to return to the original preupgraded configuration.
Override Version Stored In System Profile	Ignores the Red Hat Network profile associated with a previous version of Red Hat Linux that may have files on this computer.
Package Storage Directory	Specifies the directory for RPMs.

NOTE *As described in Table 10.4, you can download and store the updated RPMs on your computer. Just activate the After Installation, Keep Binary Packages On Disk option. All downloaded RPMs are then saved by default in the* `/var/spool/up2date` *directory. You can use the RPMs that you've saved to install updated software on other computers. However, don't expect support from Red Hat on those other computers.*

Now click the Package Exceptions tab, shown in Figure 10.10. Here, you can specify the packages and filenames that won't be upgraded through the Red Hat Network, at least not without your approval. Note that upgraded `kernel*` RPMs are on this list.

When you've finished making Red Hat Network configuration changes, click OK.

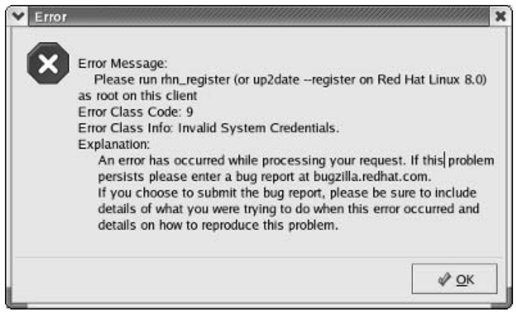
FIGURE 10.10
The do not upgrade list



Registering Your System

If you have available subscriptions (as you can see from Figure 10.7, I have two available), you're ready to register your system for the Red Hat Network. To do so, you'll want to start the Red Hat Update Agent. You can do so with the `up2date` command or by selecting Main Menu ➢ System Tools ➢ Red Hat Network. If you haven't yet registered this particular system, you'll get the error message shown in Figure 10.11. If you don't see this figure, skip to the following section.

Figure 10.11
Time to register



As you can see, you're told to register your system with the `rhn_register` command. You can execute this from a GUI or a text command line; I'm using the GUI just because it's easier to illustrate in this book. The process differs slightly if you're working from a text command line.

1. Log into the GUI as the root user (or one with administrative privileges that you configured in Chapter 9).
2. Open a command-line interface. Right-click the desktop, and select New Terminal from the pop-up menu that appears.
3. At the command line, start the registration process with the `rhn_register`command. When you see the Welcome To The Red Hat Update Agent screen, click Forward.
   A. In text-mode only, if your system is already registered, you may see a message to that effect. If you want to continue the registration process, select OK to continue.
4. Review the Red Hat privacy statement. If you're satisfied with the conditions, select Next to continue.
5. Log into the Red Hat Network using your assigned account. You'll also need to use the same e-mail address associated with your account, as shown in Figure 10.12.
6. Next, review the information about your computer that `rhn_register` is about to send to the Red Hat Network. A sample from my computer is shown in Figure 10.13. Make any appropriate changes, and select Forward to continue. You don't have to include the hardware or network profile for your computer.

FIGURE 10.12
Signing into an official Red Hat Enterprise account

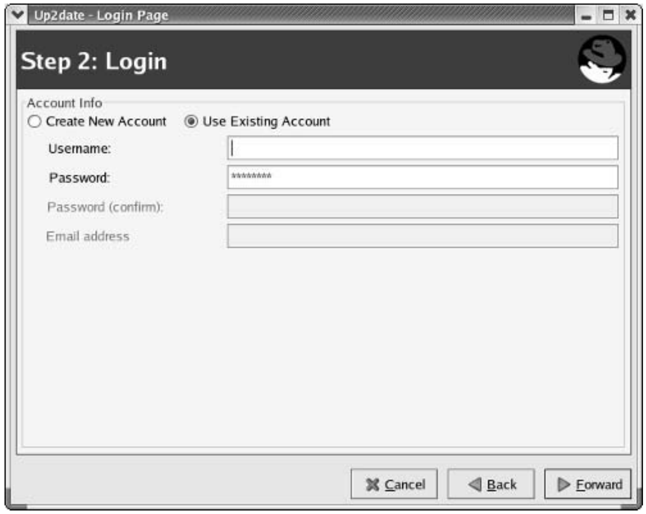
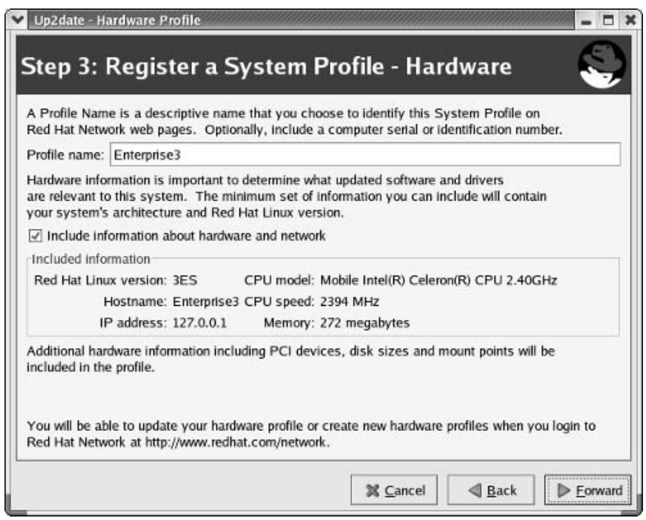
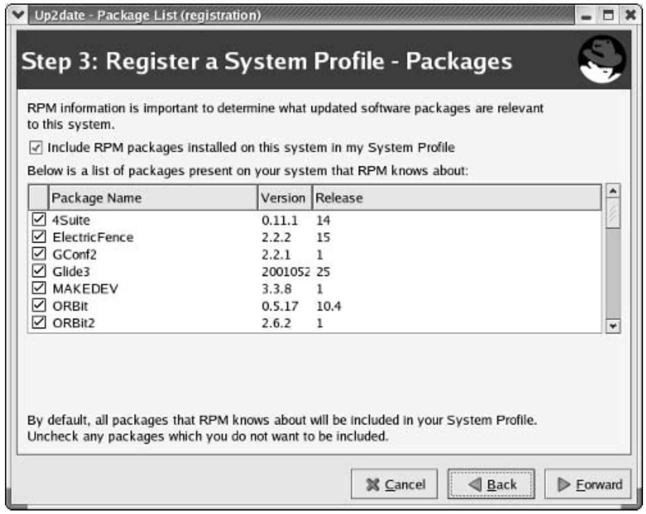


FIGURE 10.13
rhn_register collects basic information.



7. The `rhn_register` tool begins to collect a list of your current packages. While you don't have to send this list to the Red Hat Network, it is the only way Red Hat knows what RPMs are out-of-date on your computer. A sample from my computer is shown in Figure 10.14. Make any desired changes, and click Forward to continue.

FIGURE 10.14
rhn_register collects RPMs.



8. When you're ready, click Forward again to send your computer profile to the Red Hat Network. You'll see the Channels window, shown in Figure 10.16 in the next section.

Now you're ready to use `up2date` to update your system.

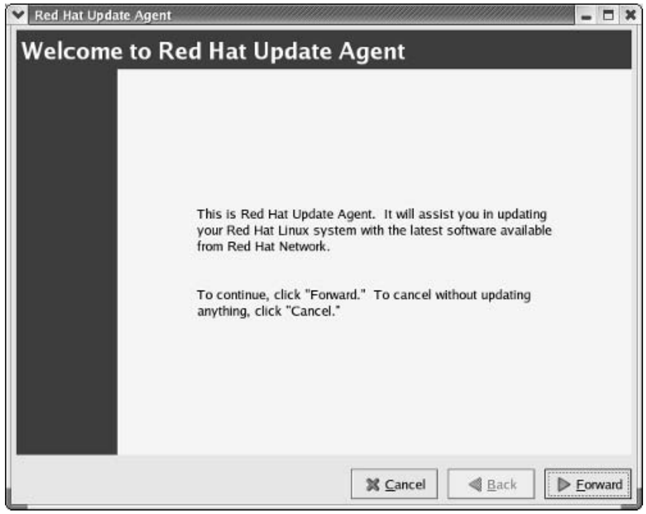
A Special Agent: *up2date*

As we've described earlier, the easiest way to keep your Red Hat Enterprise Linux system up-to-date is the Red Hat Update Agent, also known by its text command, `up2date`. When you registered your system, you sent a list of installed RPM packages to the Red Hat Network. With the Red Hat Update Agent, you can check the Red Hat database for newer RPM packages and have them installed as needed.

You can start the Red Hat Update Agent by following these steps:

1. Log into the GUI as the root user (or one with administrative privileges that you configured in Chapter 9).
2. Open a command-line interface. Right-click the desktop, and select New Terminal from the pop-up menu that appears.
3. Run the `up2date` command at the command-line interface. (You can also select Main Menu ➢ System Tools ➢ Red Hat Network.) Figure 10.15 assumes you've already registered your computer on the Red Hat Network and started `up2date` in a GUI.
4. Click Forward to continue. If you see an error message, go back to the previous section. You'll have to register your system with the Red Hat Network before proceeding.

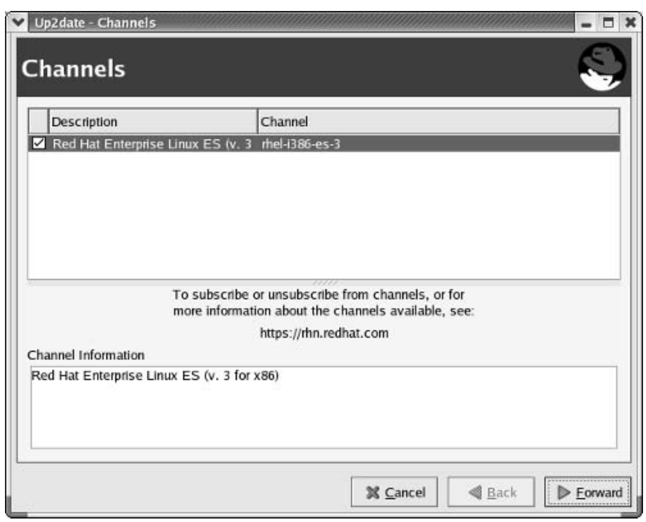
FIGURE 10.15
Starting the Red Hat Update Agent



NOTE *If you want to run the Red Hat Update Agent from a text-mode interface, you can do so with the* `up2date -u` *command. It goes through the entire process you see in this section, including RPM package updates, automatically.*

5. Assuming you've already registered, you should now see the Channels dialog box, as shown in Figure 10.16. As you can see, this dialog box lists the channel for Red Hat Enterprise Linux 3. Click Forward to continue.

FIGURE 10.16
Red Hat update channels



6. Next, `up2date` reviews the available RPM packages on the Red Hat network and compares them to what you have installed.

   A. If you're working from the original version of Red Hat Enterprise Linux 3, you'll be given a chance to upgrade to a new version of `up2date`, as shown in Figure 10.17. After you click Yes, you'll have to enable the update, as shown in Figure 10.18.

   B. Once you select the `up2date` RPM, you can click Forward to continue. Follow the prompts to download, install the new `up2date` RPM, and then restart `up2date`.

FIGURE 10.17 Upgrading *up2date*

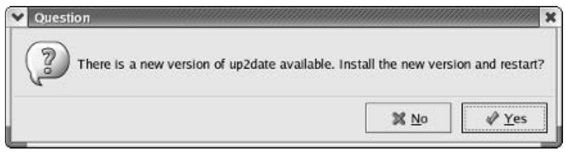
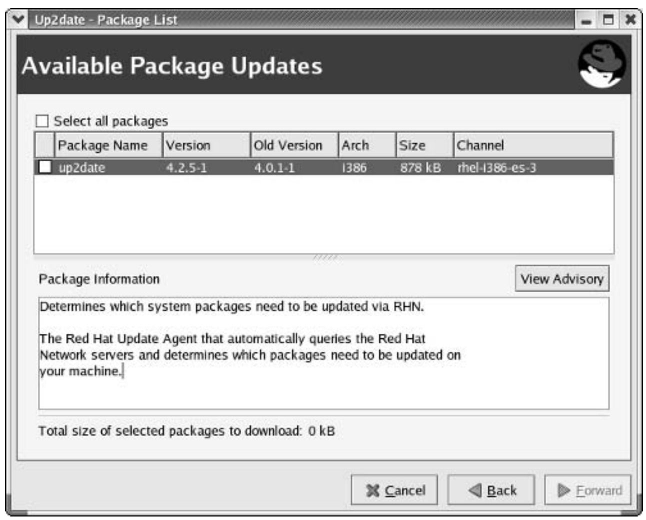


FIGURE 10.18 Authorizing a new *up2date*

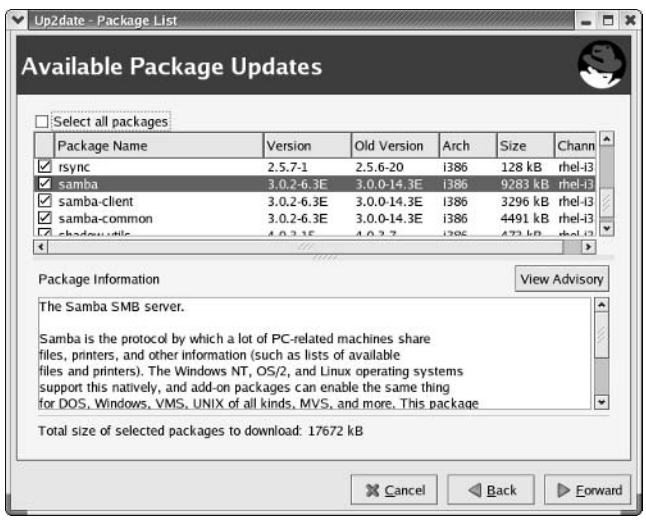


7. Assuming you've designated RPM packages to be skipped during the registration process, and there are updates of those packages available, you should see something similar to Figure 10.19. This gives you a chance to install these packages. Make any desired choices, and then click Forward to continue.

8. The next step allows you to review available updates—in other words, newer RPM packages you may install. Figure 10.20 configures the update of the Samba RPMs and more (your choices will probably not be identical). Make your selections, and then click Forward to continue.

FIGURE 10.19
Packages flagged to be skipped

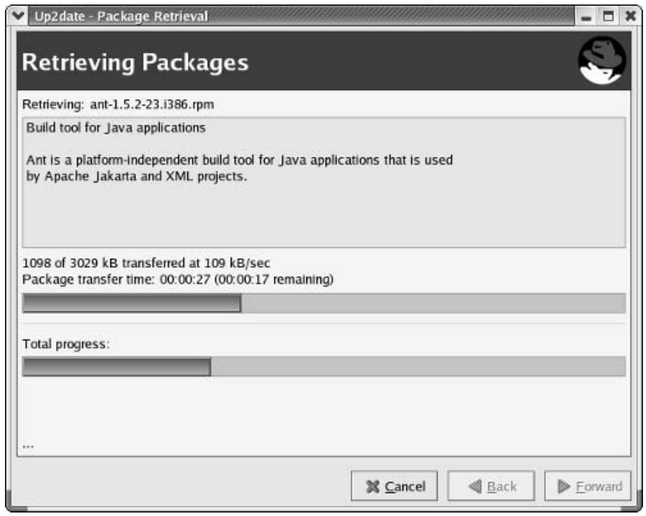
FIGURE 10.20
Available package updates



9. As described earlier, there can be RPM dependencies. If other packages need to be installed, `up2date` lists them for you. Depending on the number of packages being updated, this process may take several minutes. If there are dependencies, you'll get a chance to confirm the selection of additional RPMs. If they are acceptable, click Forward to continue. Alternatively, you can go back and deselect the packages that cause these dependencies.

10. The Red Hat Update Agent will begin downloading the desired and dependent RPM packages, similar to what is shown in Figure 10.21. The time required for the download depends on the speed of your Internet connection, as you may be downloading several hundred MB of information. Once the downloads are complete, you'll see the following message at the bottom of the Retrieving Packages screen: "All finished. Click 'Forward' to continue." Follow the prompt.

FIGURE 10.21
RPM package retrieval



11. Now that the packages are on your Linux computer, `up2date` will begin installing them. Depending on the number of packages, it may be several minutes before the process begins. Once `up2date` starts installing packages, your system should look similar to Figure 10.22. Once installation is complete, you'll see a message at the bottom of the screen that installation is "All finished. Click 'Forward' to continue." Follow the prompt.

12. Finally, you'll see a dialog box that lists the RPM packages that `up2date` installed on your computer. When you're done reviewing this list, click Finish.

Network Alert Notification

You can set up the Red Hat Network Alert Notification Tool to automatically monitor the Red Hat Network and tell you if there are critical software updates required for your system. To set this up, right-click the circular icon on your GUI taskbar. Depending on whether there are updates pending, it may be an exclamation point inside a red circle  or a blue check mark .

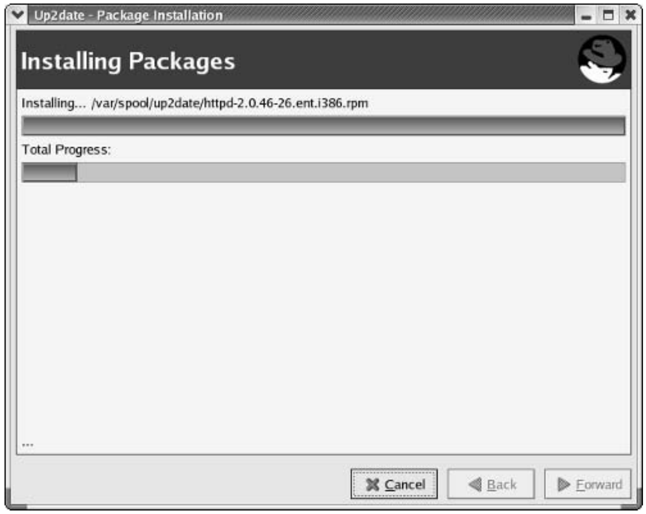


FIGURE 10.22
RPM package installation

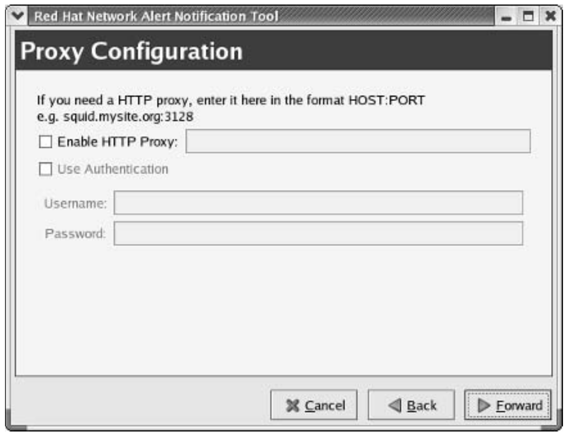
To set up this tool, follow these steps:

1. Open the GUI on your Linux computer.
2. Right-click the Red Hat Network icon in a GUI taskbar. In the pop-up menu that appears, click Configuration.
3. This opens the Red Hat Network Alert Notification Tool, as shown in Figure 10.23. Click Forward to continue.

FIGURE 10.23
Red Hat Network Alert Notification Tool

4. Review the Terms Of Service window. If it's acceptable to you, click Forward to continue.
5. If you have a proxy server that governs Internet connections from computers from your network, enter the associated information in the Proxy Configuration window shown in Figure 10.24. Once complete, click Forward to continue.

FIGURE 10.24
Proxy configuration



6. Assuming you're satisfied with your configuration, click Apply in the next window. This tool now monitors the Red Hat Network for you.

Now the tool works as a hover button. When you hover the mouse pointer over the tool, it gives you the number of packages that need to be updated, as shown in Figure 10.25.

FIGURE 10.25
The Red Hat Network Alert Notification Tool works.

Fedora Updates

When Red Hat develops a newer version of an RPM package, it may test it on the latest Fedora project distribution. The development packages formerly known as *Rawhide* are now stored in Fedora development directories. You can find these packages on Fedora download servers listed at `fedora.redhat.com/download/mirrors.html`.

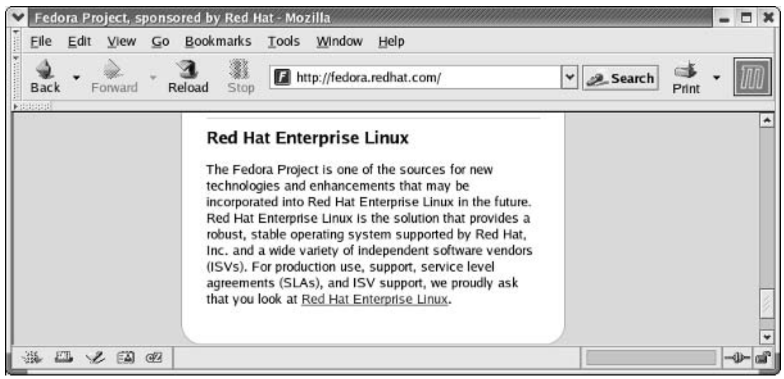
When you install a Fedora RPM on your Red Hat Enterprise Linux computer, you may be installing software developed by Red Hat. Unless and until Red Hat has incorporated this software in its enterprise distributions, such software is not supported under your Red Hat Enterprise Linux subscription.

If you're using one of the rebuilds, Red Hat support may not matter to you. As long as you understand that Fedora RPMs change frequently, it may make sense to install some Fedora software on your enterprise system on a limited basis.

There is a very active community of users and developers supporting Fedora. As of this writing, there are over 20 mailing lists and several IRC chat channels are associated with this ever-changing distribution. For the latest list, see `fedora.redhat.com/participate/communicate`. Just be warned; I get several hundred messages a day from my Fedora mailing list subscriptions. There are also ever-changing documentation resources for Fedora; three sites are `fedora.redhat.com`, `www.fedoranews.org`, and `www.fedorazine.com`.

However, as you can see at the bottom of the Fedora website shown in Figure 10.26, Red Hat Enterprise Linux is designed to be stable. As the Fedora distribution changes two or three times each year, Fedora is less than stable.

FIGURE 10.26
Enterprise note on the Fedora website



Rebuild Distribution Servers

If you're using one of the rebuilds, be careful. They haven't been able to use 100 percent of the software associated with Red Hat Enterprise Linux. For example, they don't have access to the few proprietary packages released by Red Hat. At times, they've had to use other programming libraries to rebuild some RPMs. Also, they've had to remove all Red Hat symbols and icons so they don't infringe on the Red Hat trademark.

That factor aside, they have been conscientious (so far) with respect to updates released by Red Hat. Anyone can download and install the binary RPMs released by the rebuild groups. You can download and install these updates using the `yum` RPM software.

As of this writing, at least the cAos rebuild has integrated `yum` into `up2date`. In other words, if you've installed the cAos CentOS-3 rebuild on your computer, you can use its version of `up2date` to update the RPMs on your system. They've set up their own repositories of rebuilt updated Red Hat Enterprise Linux 3 RPMs. The CentOS-3 rebuild version of `up2date` downloads and installs these packages; the process is essentially the same as the `up2date` process described earlier in this chapter.

NOTE *CentOS-3 is the cAos project's rebuild of Red Hat Enterprise Linux 3. For more information, see* `www.caosity.org`.

Older Versions of Red Hat

Red Hat Enterprise Linux 3 was developed from Red Hat Linux 9. Based on the original versions, more than 250 RPMs are identical on both systems. A number of RPMs are close enough to make no difference even to most Linux system administrators.

In other words, a great deal of software that's included with Red Hat Linux 9 is easily compatibly with Red Hat Enterprise Linux 3. That has allowed me to load some software such as Red Hat Linux 9's `anacron` (Chapter 13) on Red Hat Enterprise Linux 3.

The *yum* Alternative

Two packages other than `up2date` can help you keep your RPMs updated to the latest requirements. They are `yum` and `apt`. `yum` is the Yellow Dog Updater, Modified. Yellow Dog Linux is a distribution designed for the PowerPC CPU; in other words, you can install it on Apple PowerPC computers. `apt` is the Advanced Package Tool. Neither package is included with or supported by Red Hat Enterprise Linux 3. However, if you're using one of the rebuilds, you should configure one of these packages to keep your system up-to-date.

There are good reasons to use each of these packages. In principle, we prefer `yum`, as it was created originally for RPMs. While the roots of `apt` are in Debian, which argues for its stability, it was originally designed for and includes software to handle `deb` software packages—and retains that extra overhead. One handy place where you can download both the `apt` and `yum` packages is `apt.freshrpms.net`; versions for Red Hat Linux 9 and Fedora Linux 1 (and more) are available as of this writing. Just don't install both `apt` and `yum`, so the associated configuration and log files can properly keep your system up-to-date.

yum as Configured

As described earlier, cAos has already integrated `yum` into its version of `up2date`, so no additional work is required for its CentOS-3 rebuild. The folks behind White Box Enterprise Linux have done the same thing. Other rebuilds may customize their own version of `yum`. Please refer to their release notes for detailed information.

Once configured, you can use `yum` to use your connection to your update servers in a number of ways. We've listed some of the available commands are listed in Table 10.5. The first time you run one of these commands, the command refers to the server(s) you've configured in `/etc/yum.conf` to download applicable headers to the related log file. For example, the cAos version downloads them to the `/var/cache/yum` directory.

Table 10.5: Some *yum* Command Options

Command	Description
`yum list`	Downloads and lists available headers, with package version numbers.
`yum check-updates`	Checks your RPMs against the current lists for any needed upgrades.
`yum install` ***packagename***	Installs the RPM of your choice; if already installed, this allows you to update that package.

As an example, we've run the `yum list | less` command in Figure 10.27. It looks through the servers listed in `/etc/yum.conf`, and it returns package names, associated CPU architecture, version number, and server.

FIGURE 10.27
A `yum` list of packages

```
Gathering header information file(s) from server(s)
Server: CentOS-3build7 - Addons
Server: CentOS-3build7 - Base
Server: CentOS-3build7 - Extras
Server: CentOS-3build7 - Testing
Server: CentOS-3build7 - Updates
Finding updated packages
Downloading needed headers
Name                                Arch   Version                  Repo
--------------------------------------------------------------------------------
Canna                               i386   3.6-20                   base
Canna-devel                         i386   3.6-20                   addons
Canna-libs                          i386   3.6-20                   base
FreeWnn                             i386   1.11-36                  base
FreeWnn-common                      i386   1.11-36                  base
FreeWnn-devel                       i386   1.11-36                  addons
FreeWnn-libs                        i386   1.11-36                  base
GConf2-devel                        i386   2.2.1-1                  base
Glide3-devel                        i386   20010520-25              addons
Gtk-Perl                            i386   0.7008-31                addons
ImageMagick-c++                     i386   5.5.6-4                  addons
ImageMagick-c++-devel               i386   5.5.6-4                  addons
ImageMagick-devel                   i386   5.5.6-4                  addons
:
```

Once you have the headers on your system, you can use them to check the server for any more up-to-date packages. I've done this in Figure 10.28 on a CentOS-3 system with the `yum check-updates` command.

FIGURE 10.28
Checking `yum` headers for updates

```
[root@localhost root]# yum check-update
Gathering header information file(s) from server(s)
Server: CentOS-3build7 - Addons
Server: CentOS-3build7 - Base
Server: CentOS-3build7 - Extras
Server: CentOS-3build7 - Testing
Server: CentOS-3build7 - Updates
Finding updated packages
Downloading needed headers
Name                                Arch   Version                  Repo
--------------------------------------------------------------------------------
centos-yumconf                      noarch 1-5                      update
gdk-pixbuf                          i386   1:0.22.0-6.1.1           update
libxml2                             i386   2.5.10-6                 update
libxml2-devel                       i386   2.5.10-6                 update
libxml2-python                      i386   2.5.10-6                 update
mod_python                          i386   3.0.3-3.ent              update
nfs-utils                           i386   1.0.6-7.EL               update

[root@localhost root]#
```

You can update your system as desired using the `yum install` ***`packagename`*** command. You can also configure `yum` to point to your own local update servers. You can find other `yum` repositories at the Yum website at `linux.duke.edu/projects/yum/repos`.

SETTING UP YOUR OWN *YUM* REPOSITORIES

There is one possible problem with these `yum` repositories; they're outside your network. If you're working in the enterprise, you may not want all of the computers on your network to *simultaneously*

access the GB of files from an outside server. If you have enough Red Hat Enterprise Linux systems on your network, it may make sense to set up your own `yum` repository.

This is a straightforward process; the basics are included in the `yum` HOWTO created at Duke University at `www.phy.duke.edu/~rgb/General/yum_HOWTO/yum_HOWTO/`.

As an example, assume that you've installed the original version of Red Hat Enterprise Linux 3 on your computers. You've just downloaded the updates CD and want to set it up as a `yum` repository on your network. Follow these basic steps:

1. Copy the desired files and directories to a location associated with the server that you'll use, such as `/var/ftp/pub` for an FTP server or `/var/www/html` for a web server.

2. Set up headers in the directory with the updates using the `yum-arch` command. For example, the following command collects headers in the `/var/www/html/inst/Updates/headers` directory with the following command:

   ```
   # yum-arch /var/www/html/inst/Updates
   ```

3. Start or restart the appropriate server; in this case, it's the Apache web server. For more information on Apache, read Chapter 25. Make sure any firewall you've configured allows traffic to and from this server.

4. Configure `/etc/yum.conf` on the clients on your network. If your server name is `server.redhat.com`, you'd add the following stanza to your `yum.conf` file (you can substitute the IP address for the server name):

   ```
   [RHELUpdate]
   name=RHEL-Updates
   baseurl=http://server.redhat.com/inst/Updates
   ```

The next time you run a command such as `yum list`, you can watch as it downloads the headers from your new local `yum` repository. It's now ready for use.

Automating *yum*

The standard `yum` service may not be started by default. You can configure it to start automatically with the `chkconfig` command. When you do, it'll automatically run the associated `cron` job on a daily basis. We describe both systems in Chapter 13. Fortunately, the commands in the standard `/etc/cron.daily/yum.cron` job are straightforward. The following command essentially checks to see if the `yum` service is running:

```
if [ -f /var/lock/subsys/yum ]; then
```

The following commands first looks for any installs updated `yum` package and then runs through and installs any available updates:

```
/usr/bin/yum -R 10 -e 0 -d 0 -y update yum
/usr/bin/yum -R 120 -e 0 -d 0 -y update
```

For more information on the switches shown in the `yum` commands, refer to the `yum` man page.

Summary

If you want to install new software in Red Hat Enterprise Linux, you need to know how to manage RPM packages. You can use the `rpm` command to upgrade or install new packages locally from a source such as a CD or remotely from a FTP or HTTP server.

The `rpm` command is flexible. With the right switches, you can query the status, the list of files, or even the ownership of a package. A properly configured RPM package lists dependencies. For example, if you need the GNU C Compiler for something such as the Linux kernel source, the `rpm` command won't let you install the `kernel-source` package first, at least not by default. If you need to find the right RPM, the `rpmdb-redhat-*` RPM package provides a database of all RPMs associated with your current Red Hat Enterprise Linux distribution.

But the `rpm` command can't work with entire package groups. That's where the Red Hat GUI Package Management tool can help. It helps you work with many of the same package groups you may have configured during the graphical installation process.

Linux is associated with easy accessibility to the source code. Red Hat Enterprise Linux supports this with source RPMs. Once you've installed the `rpm-build` package, you can use the `rpmbuild` command to create binary RPMs from the source. All you need is a properly configured `.spec` file.

Spec files are included with Red Hat Enterprise Linux SRPMs, and you can modify them to meet your needs. Alternatively, you can create your own `.spec` file to create a binary RPM from a tarball package.

It may be a bit too easy to become dependent on RPMs and ignore security issues. Therefore, Red Hat Enterprise Linux supports the Pretty Good Privacy system. All you need is a genuine `RPM-GPG-KEY` file, available from several sources. Then you can verify the integrity of any RPM package with the `rpm -K` *`packagename`* command. If you suspect a problem with a specific file or command, you can even verify the integrity of that specific file with the `rpm -Vf` *`filename`* command.

If you're looking for the latest RPMs, use the Red Hat Update Agent. On the official version of Red Hat Enterprise Linux, it allows you to download updates of RPM packages as needed. On some of the "rebuilds," the Update Agent is linked to `yum` to download and install updates from third-party repositories. They may also include their own customized version of `yum` for updates.

In the next chapter, you'll learn to analyze the boot process in detail. As you learn about the Linux boot process, you'll gain skills that can help you troubleshoot various kinds of boot problems. Finally, you can use the Red Hat installation CD's `linux rescue` mode to get around most boot problems so you can repair any damaged files.

Chapter 11

Configuring and Troubleshooting the Boot Process

SOMEDAY, RED HAT ENTERPRISE Linux may have problems booting on your computer. If you see a message such as `kernel panic`, don't panic! You may not even have to restore your system from a backup. If you know the basic boot configuration files, you can quickly and easily diagnose and solve most boot problems.

To understand how Linux boot configuration files work, you need to understand the basic boot process, from hardware detection through runlevel management.

Then you can get into the nitty-gritty of the key boot configuration files for managing hardware, for finding your kernel, for starting your terminals, and for initializing services at the appropriate runlevel.

If you have a problem, you can create a boot disk that will normally get you around most problems. Otherwise, the Red Hat Enterprise Linux installation boot disk, even the first installation CD, can offer you a rich variety of rescue modes. This chapter covers the following topics:

- Exploring the basic boot process
- Understanding the default configuration files
- Troubleshooting and using rescue disks

Exploring the Basic Boot Process

Before getting into the nitty-gritty of Red Hat Enterprise Linux configuration files, it's important to have a "big picture" overview of the process. While small changes can keep Red Hat Enterprise Linux from booting, an understanding of the big picture can help you identify the problem quickly.

When you start your Linux computer, several basic steps are involved in the process. Hardware is initialized through your Basic Input/Output System (BIOS). The BIOS points to the Linux bootloader. Once the bootloader starts, it opens the kernel. Next, it starts `init`, the so-called "first program," which then loads your kernel, and it then moves to initialize other startup programs. Finally, Linux finds the default runlevel and starts all associated processes.

We provide detailed information on each of these processes later in this chapter.

Initializing Hardware

While this is not a book on computer hardware, it's helpful to know some basics. Then it's easier to determine if you have a hardware problem or a Linux problem.

Everything on a standard PC starts with the BIOS. The first step, associated with a series of beeps, is known as the POST (power-on self-test), which checks connections to basic hardware. It looks for other BIOSs related to IDE and SCSI hard drives. It may also detect other basic hardware on your system.

TIP *If you're interested in the Linux+ certification exam from CompTIA, you need to know a lot more about PC hardware. Do note that the CompTIA Linux+ exam requirements will change around the end of 2004. For more information on Linux+ exam requirements, see Chapter 27. For more information on PC hardware, see the* Complete PC Upgrade and Maintenance Guide, Fifteenth Edition *(Sybex)*.

After Linux initiates the loading process through the bootloader, it begins to detect hardware using the `kudzu` utility. Then it adds modules related to your hardware, using settings stored in `/etc/modules.conf`. You can analyze the results with the `dmesg` command. If you're having a hardware problem, a little detective work with `dmesg` output can help you identify the trouble.

Bootloaders

There are two basic Linux bootloaders, the Grand Unified Bootloader (GRUB) and the Linux Loader (LILO). GRUB is the default for Red Hat Enterprise Linux. LILO is now obsolete and will probably be removed in some future release of Red Hat Enterprise Linux.

In either case, the bootloader is used for the following four purposes:

- To select an operating system (if more than one is installed on your computer)
- To identify the partition with the appropriate boot files
- To locate the kernel
- To run the Initial RAM disk to set up the kernel and associated modules

Runlevels

A *runlevel* is a specific way to organize initialized software in Linux. Different services are started and stopped at different runlevels. When you start Red Hat Enterprise Linux, it looks to `/etc/inittab` to determine the default runlevel, which then points to an associated subdirectory of `/etc/rc.d` to identify the services to kill and start. We'll describe the six default Red Hat runlevels shortly.

Understanding the Default Configuration Files

To recap, there are key startup configuration files for hardware, for the bootloader, and for runlevels. The hardware configuration files help you determine what was detected. The bootloader enables you to trace the location of the kernel, the Initial RAM disk, and any other operating systems on your computer. The directories for each runlevel help you customize the processes that start and stop on your Linux computer.

Hardware Detection

Once GRUB or LILO finds your boot files, the next step is to make a connection between the Linux kernel and your computer's hardware. The Linux hardware detection process consists of several parts. First, Linux takes data related to basic hardware from your BIOS. Next, it uses the `kudzu` utility to look for new hardware on your system. Assuming you have a default modular kernel, it then inserts any modules related to specialized hardware from the `/etc/modules.conf` file. You can inspect the messages related to this process with the `dmesg` command.

Kernel Connections

The `dmesg` command should show you how your kernel interacts with your hardware as Linux starts on your computer. It starts with your BIOS; uses related information to find your CPU, hard drives, PCI (Peripheral Component Interconnect) devices, and communications ports; starts the appropriate filesystems on the right partitions; and finally configures other basic devices related to keyboards and mice. Figure 11.1 shows an excerpt from my `dmesg` output.

Figure 11.1
Excerpt from *dmesg*

```
Linux version 2.4.21-4.EL (bhcompile@daffy.perf.redhat.com) (gcc version 3.2.3 2
0030502 (Red Hat Linux 3.2.3-20)) #1 Fri Oct 3 18:13:58 EDT 2003
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)
 BIOS-e820: 00000000000dc000 - 00000000000e0000 (reserved)
 BIOS-e820: 00000000000e4000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 00000000106f0000 (usable)
 BIOS-e820: 00000000106f0000 - 00000000106fc000 (ACPI data)
 BIOS-e820: 00000000106fc000 - 0000000010700000 (ACPI NVS)
 BIOS-e820: 0000000010700000 - 0000000010800000 (usable)
 BIOS-e820: 00000000fec00000 - 00000000fec10000 (reserved)
 BIOS-e820: 00000000fee00000 - 00000000fee01000 (reserved)
 BIOS-e820: 00000000fffe0000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
264MB LOWMEM available.
On node 0 totalpages: 67584
zone(0): 4096 pages.
zone(1): 63488 pages.
zone(2): 0 pages.
Kernel command line: ro root=LABEL=/
Initializing CPU#0
:
```

From the sample output, you can identify one CPU and 264MB of memory. If you actually have more than one CPU and additional RAM installed, this output tells you that Linux did not detect this additional hardware.

Kudzu

The current version of Kudzu is the culmination of Linux efforts to support plug-and-play hardware. In the past, using plug-and-play hardware on Linux was at best an uncertain venture. Now it manages any new hardware that you throw at it without a hitch.

Kudzu works by looking at the various ports on your computer. If it detects and recognizes new hardware, it adds the relevant information, such as device and driver names, to `/etc/sysconfig/hwconf`.

If special hardware drivers are required, specifications are added to `/etc/modules.conf`. Linux reads this file during the boot process to load the required drivers the next time you start your computer.

If you've just added new hardware and want to make sure Red Hat Enterprise Linux detects it properly, just run the `kudzu` command. If additional configuration is required, you could be taken to

a text version of `redhat-config-mouse` (see Chapter 2); the steps look similar to the deprecated `mouseconfig` utility. In rare cases, you may be prompted to add information such as IRQ ports, I/O addresses, or DMA channels.

Kernels and Hardware

Linux makes it easy to see how the Linux kernel views your hardware. Just look in the `/proc` directory. As shown in Table 11.1, various files in `/proc` can give you additional information on the hardware that is connected to Red Hat Enterprise Linux.

Table 11.1: Selected Hardware Files in */proc*

File	Description
`apm`	Advanced power management battery status
`cpuinfo`	Detected CPUs
`dma`	Assigned DMAs
`ide`	Directory specifying attached IDE devices
`interrupts`	Assigned IRQs
`ioports`	Assigned I/O addresses
`modules`	Installed driver modules; same as `lsmod` output
`partitions`	Basic partition information
`pci`	Detected PCI devices
`scsi`	Directory specifying attached SCSI devices

The information is quite detailed. For example, take a look at the `/proc/cpuinfo` file in Figure 11.2. Not only does it show the rated and effective speed of the CPU, but it also shows the `cpu family`, which helps you find the optimized Linux kernel to use for your system. In this case, I'd use the `kernel-`*`versionnumber`*`.i686.rpm` package. You'll see how this helps in the next chapter.

The */etc/modules.conf* Settings

Sometimes Red Hat Enterprise Linux needs a little help with kernel configuration settings. Sometimes default plug-and-play settings for different components interfere with each other. That's where the `/etc/modules.conf` configuration file steps in. It's where Linux stores driver, device, and address settings for various hardware components. Take the following excerpt from my `/etc/modules.conf` file:

```
post-install sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -L
➥ >/dev/null 2>&1 || :
pre-remove sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -S
➥ >/dev/null 2>&1 || :
alias eth0 natsemi
alias usb-controller usb-ohci
```

FIGURE 11.2
Kernel information on the CPU

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Mobile Intel(R) Celeron(R) CPU 2.40GHz
stepping        : 8
cpu MHz         : 2395.073
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
bogomips        : 4639.94
~
~
~
"/proc/cpuinfo" [readonly] 19L, 429C                        4,7-16        All
```

Note how this defines drivers for a sound card, the first Ethernet card, and the USB controller. As Linux detects more and more hardware, the importance of this file will decline over time.

When you change the `/etc/modules.conf` file, you can test the results immediately. For example, if your computer includes a Sound Blaster card, you can test any settings you change with the following command:

```
# modprobe sb
```

If you don't see an error message, check the `lsmod` command again. Your sound card is probably now installed. Otherwise, check the error messages carefully for clues on your next step, which is probably to try different hardware settings.

Listing Modules

You can verify whether Red Hat was able to detect your hardware. Besides reviewing the earlier discussion on the `/proc` directory, you can review installed modules with the `lsmod` command. For example, this command on my computer lists a series of modules in Figure 11.3.

As you can see, each module has a file size in bytes. Some modules depend on others; for example, note how the `mii` module is required for the `pcnet32` network driver module. In other words, if you tried to remove the `mii` module with the following command, you'd get an error message:

```
# rmmod mii
mii: Device or resource busy
```

If you remove the `pcnet32` network card from your computer, Linux won't install either module the next time you start your computer. Alternatively, you could deactivate your network cards (`ifconfig eth0 down`) and then remove the modules in order:

```
# rmmod pcnet32
# rmmod mii
```

FIGURE 11.3
lsmod lists installed modules.

```
Module                  Size  Used by    Not tainted
smbfs                  44528   1  (autoclean)
udf                    98464   0  (autoclean)
ide-cd                 35680   0  (autoclean)
cdrom                  33696   0  (autoclean) [ide-cd]
nfsd                   85456   8  (autoclean)
lockd                  59856   1  (autoclean) [nfsd]
sunrpc                 85692   1  (autoclean) [nfsd lockd]
parport_pc             19076   1  (autoclean)
lp                      9028   0  (autoclean)
parport                37088   1  (autoclean) [parport_pc lp]
autofs                 13364   0  (autoclean) (unused)
iptable_filter          2412   0  (autoclean) (unused)
ip_tables              15776   1  [iptable_filter]
pcnet32                18080   1
mii                     3976   0  [pcnet32]
crc32                   3712   0  [pcnet32]
floppy                 58160   0  (autoclean)
microcode               4724   0  (autoclean)
keybdev                 2976   0  (unused)
mousedev                5524   1
hid                    22212   0  (unused)
input                   5888   0  [keybdev mousedev hid]
usb-uhci               26412   0  (unused)
usbcore                79392   1  [hid usb-uhci]
ext3                   91592   3
jbd                    52336   3  [ext3]
raid1                  14988   2
[root@Enterprise3 root]#
```

You could install modules just as easily; for example, if you need to install a new 3Com EtherLink network card and Linux isn't detecting it, you can try installing the associated module with the following command:

```
# insmod 3c589_cs
```

If successful, you won't see any error messages; check the result with the `lsmod` command. You should see the network card module in the output.

The Bootloader

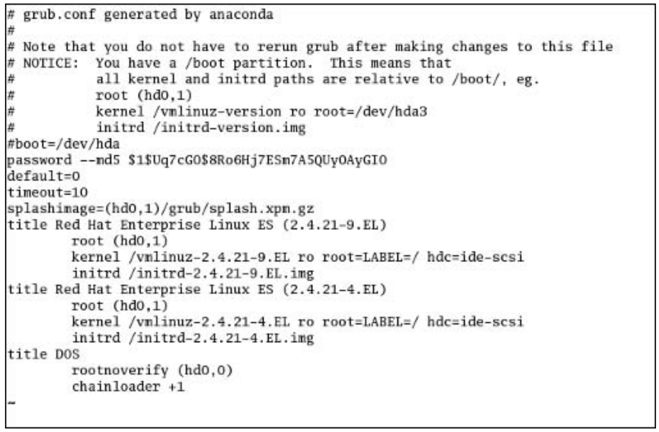
As we mentioned earlier, the default Red Hat Enterprise Linux bootloader is known as GRUB. It is a significant improvement over LILO in a number of ways, including the following:

- It can be password-protected.
- It is easy to edit during the boot process. You can try different boot parameters without permanent changes to the GRUB configuration file.
- It can boot Windows 32-bit operating systems (NT/2000/XP/2003) from the Master Boot Record area of your hard drive.
- It supports Logical Block Addressing (LBA) mode, which can help your computer find the `/boot` files, especially if they are beyond the 1024th cylinder on your hard drive.

LILO is now deprecated; Red Hat plans to remove LILO from Red Hat Enterprise Linux, probably in the next version. Therefore, this older bootloader is not covered in this book.

Take a look at a typical GRUB configuration file in Figure 11.4, from `/boot/grub/grub.conf`.

FIGURE 11.4
Dual-boot GRUB



```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,1)
#          kernel /vmlinuz-version ro root=/dev/hda3
#          initrd /initrd-version.img
#boot=/dev/hda
password --md5 $1$Uq7cGO$8Ro6Hj7ESm7A5QUyOAyGIO
default=0
timeout=10
splashimage=(hd0,1)/grub/splash.xpm.gz
title Red Hat Enterprise Linux ES (2.4.21-9.EL)
        root (hd0,1)
        kernel /vmlinuz-2.4.21-9.EL ro root=LABEL=/ hdc=ide-scsi
        initrd /initrd-2.4.21-9.EL.img
title Red Hat Enterprise Linux ES (2.4.21-4.EL)
        root (hd0,1)
        kernel /vmlinuz-2.4.21-4.EL ro root=LABEL=/ hdc=ide-scsi
        initrd /initrd-2.4.21-4.EL.img
title DOS
        rootnoverify (hd0,0)
        chainloader +1
~
```

The variables shown in Figure 11.4 are explained in Table 11.2.

TABLE 11.2: SELECTED GRUB VARIABLES

VARIABLE	COMMENT
`password`	Password-protects GRUB. With the `--md5` switch, the password can be entered in encrypted format.
`default`	Specifies the default operating system. If `default=0`, the operating system shown in the first stanza boots automatically if there is no user input.
`timeout`	Sets the time limit before GRUB starts the default, in seconds.
`splashimage`	Notes the default GRUB image.
`title`	Sets the option as shown in the GRUB menu.
`root`	Specifies the partition with the `/boot` files.
`kernel`	Notes the location of the Linux kernel.
`initrd`	Points to the location of the Initial RAM disk.
`rootnoverify`	Specifies the partition with boot files for a sensitive operating system such as Windows XP.
`chainloader`	With +1, looks for boot files in the first sector of the noted partition.

The `root` variable in GRUB may be confusing, because it actually refers to the partition with the `/boot` directory. Note the data associated with root, such as `(hd0,0)` or `(hd0,1)`. This data points to the partition with the boot files for that operating system.

If you're having trouble with hardware, use the hardware modules described earlier as much as possible. However, you may need to give Linux some help finding hardware critical to the boot process,

such as a hard drive or a SCSI controller for a hard drive. In that case, you should specify a module in the `kernel` line. For example, if you want to specify an IRQ of 9, an IO address of 0x330, and a SCSI ID of 7 for an older Adaptec controller, add the following command to the `kernel` line in your `grub.conf` configuration file:

```
kernel /vmlinuz-2.4.21-9.EL ro root=LABEL=/ aha152x=0x330,9,7
```

The boot hard disk is shown as a comment as `/dev/hda`. Therefore, `root (hd0,1)` points to Linux boot files on the first IDE hard drive, on the second partition, also known as `/dev/hda`. Similarly, the `rootnoverify (hd0,0)` setting points to DOS boot files on the first IDE hard drive, on the first partition (`/dev/hda1`).

Note *For convenience,* `/etc/grub.conf` *is linked to the actual bootloader configuration file,* `/boot/grub/grub.conf`*.*

Tip *The word* root *has several meanings in Linux. There is the root user, with a home directory of* `/root`*. There is the top-level root directory, associated with the forward slash,* `/`*. And in GRUB, the root variable actually points to the partition with the* `/boot` *directory. So when you see* `/` *in the GRUB configuration file, it's really the* `/boot` *directory.*

Adding a Password to GRUB

If you forgot to add a GRUB password during Red Hat Enterprise Linux installation, it's easy to add a secure MD5 password to GRUB. Just use the `grub-md5-crypt` command. When prompted, enter the password of your choice. You'll get a strange-looking series of characters that you can copy to the GRUB configuration file, in the format shown in Figure 11.4.

It's easy to copy this password from the command line. Just use your mouse to highlight the password. Open `/etc/grub.conf` in a text editor. Right-click your mouse in the desired location, and then Linux automatically inserts the highlighted MD5 password. Alternatively, if you're in the GNOME terminal, right-clicking opens a pop-up menu that allows you to copy and paste the highlighted text.

/etc/inittab

Linux now initializes the key files, processes, and applications on your system. The governing configuration file is `/etc/inittab`. Open it in your favorite text editor. The key variable in this file is `initdefault`; the other variables just set up important parts of the Linux environment. My `/etc/inittab` file is shown in Figure 11.5.

The `initdefault` variable sets the default runlevel, which starts when you boot Linux. For example, the following line configures your computer to start in runlevel 3.

```
id:3:initdefault
```

There are six standard Red Hat Enterprise Linux runlevels, as shown in Table 11.3 (runlevel 4 is not used by Red Hat). In the next section, we'll explore what happens when Linux boots in runlevel 3.

TABLE 11.3: STANDARD RED HAT ENTERPRISE LINUX RUNLEVELS

RUNLEVEL	FUNCTION
0	Halt; shuts down Linux; `init` stops all services currently running on your computer.
1	Single-user mode; no networking; `init` starts just the programs needed to allow one user to log into your Linux system; as you'll see later in this chapter, you can go into single-user mode to fix critical files and more.
2	Multiuser mode; no NFS access; `init` starts the programs that allow multiple users to log into your Linux system simultaneously.
3	Multiuser mode with networking; `init` starts the network daemons on your computer after the multiuser runlevel.
5	Graphical login; `init` starts your network programs and then starts X Window programs that can be split between client and server.
6	Reboot; shuts down Linux and restarts your computer at the runlevel defined by the `id` command in `/etc/inittab`.

Other Linux distributions may use different standard runlevels.

WARNING *Do not set your default runlevel to 0 or 6. If you do, your computer will either shut down or go into a continuous reboot cycle when you start Linux.*

FIGURE 11.5
/etc/inittab

```
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
```

The standard Red Hat Enterprise Linux `/etc/inittab` file includes several other important commands. The following command

```
si::sysinit:/etc/rc.d/rc.sysinit
```

runs the `rc.sysinit` script, which activates configured networks, quotas, fonts; mounts filesystems; activates Logical Volume Management (LVM) and RAID partitions; loads hardware modules; and more. In short, `rc.sysinit` sets the stage for Linux to activate services.

To help Microsoft Windows users, `/etc/inittab` associates the Ctrl+Alt+Del key combination with the `shutdown` command.

Tip *If you're setting up a Linux server, you may want to comment out the* `ca::ctrlaltdel:/sbin/shutdown -t3 r now` *command. You don't want the frustration of one user to halt the system for everyone.*

By default, Red Hat Enterprise Linux uses `/etc/inittab` to set up six virtual terminal consoles, `tty1` through `tty6`. You can access different virtual consoles by pressing Ctrl+Alt+F*n*, where *n* is the number of the console. Red Hat Enterprise Linux allows you to configure up to 12 virtual consoles, with commands such as the following in `/etc/inittab`:

```
1:2345:respawn:/sbin/mingetty tty1
```

This command configures the first virtual console (`tty1`) whenever Linux starts runlevels 2, 3, 4, or 5.

Tip *If you've just edited* `/etc/inittab`*, you may not need to reboot. For example, if you've added a virtual console, the* `telinit q` *command forces Linux to reread* `/etc/inittab`.

The First Process: *init*

Closely related to `/etc/inittab` is the first process, `init`. It works at several different runlevels, primarily scripts in the `/etc/rc.d` directory. For example, if you run the `init 5` command, Linux runs the scripts in the `/etc/rc.d/rc5.d` directory.

Starting a Runlevel

Now we'll look at how Red Hat Enterprise Linux starts a runlevel with the `initdefault` variable. As we described earlier, it's common for Red Hat Enterprise Linux to start in runlevel 3, full multiuser mode. When Linux reads the desired runlevel, it starts the associated script. In this case, the following command starts all of the scripts associated with runlevel 3:

```
l3:3:wait:/etc/rc.d/rc 3
```

This command points to a set of scripts at the associated runlevel, and it then executes kill and start scripts, in that order. It's easy to compare two different runlevels. Just examine the list of scripts in the appropriate directories. Figure 11.6 compares the scripts from runlevel 3 with runlevel 1.

Note *The kill and start scripts you see on your computer vary with the services that you've installed and those that you've activated in that runlevel.*

FIGURE 11.6
Resource control scripts

```
[root@Enterprise3 root]# \ls /etc/rc.d/rc3.d/
K05innd             K35vncserver  K74nscd            S17keytable     S85gpm
K05saslauthd        K35winbind    K74ypserv          S20random       S85httpd
K10psacct           K36lisa       K74ypxfrd          S24pcmcia       S90crond
K15dc_client        K40smartd     K87portmap         S25netfs        S90squid
K15dc_server        K45named      K92iptables        S26apmd         S90xfs
K20netdump-server   K50netdump    S00microcode_ctl   S28autofs       S91smb
K20nfs              K50snmpd      S05kudzu           S55cups         S95atd
K20rwhod            K50snmptrapd  S08arptables_jf    S55sshd         S97rhnsd
K20spamassassin     K50tux        S08ip6tables       S56rawdevices   S99local
K24irda             K50vsftpd     S10network         S56xinetd       S99mdmonitor
K34dhcrelay         K70aep1000    S12syslog          S58ntpd         S99mdmpd
K34yppasswdd        K70bcm5820    S13irqbalance      S59hpoj
K35dhcpd            K73ypbind     S14nfslock         S80sendmail
[root@Enterprise3 root]# \ls /etc/rc.d/rc1.d/
K03rhnsd            K25squid       K50tux        K86nfslock
K05atd              K25sshd        K50vsftpd     K87irqbalance
K05innd             K30sendmail    K50xinetd     K87portmap
K05saslauthd        K34dhcrelay    K60crond      K88syslog
K10cups             K34yppasswdd   K61hpoj       K90network
K10psacct           K35dhcpd       K70aep1000    K92arptables_jf
K10xfs              K35smb         K70bcm5820    K92ip6tables
K15dc_client        K35vncserver   K72autofs     K92iptables
K15dc_server        K35winbind     K73ypbind     K95kudzu
K15gpm              K36lisa        K74apmd       K96pcmcia
K15httpd            K40smartd      K74nscd       K99mdmonitor
K20netdump-server   K44rawdevices  K74ntpd       K99mdmpd
K20nfs              K45named       K74ypserv     K99microcode_ctl
K20rwhod            K50netdump     K74ypxfrd     S00single
K20spamassassin     K50snmpd       K75netfs      S17keytable
K24irda             K50snmptrapd   K80random
[root@Enterprise3 root]#
```

The directories are fairly straightforward; kill scripts start with a *K*, while start scripts begin with an *S*. These scripts execute in the order shown. But differences do exist. In this configuration, runlevel 3 starts more than 25 services, many related to networking. Runlevel 1 kills just about every available service, except the two needed for single-user mode. No networking or multiuser configurations are required in single-user mode.

NOTE *Remember, a script such as* `S05kudzu` *starts a service, and a script such as* `K15httpd` *kills a different service. For more information on service management, see Chapter 13.*

Troubleshooting and Using Rescue Disks

As a system administrator, you'll need to examine and edit a number of configuration files. When changes are made, mistakes are possible. For example, if you make a mistake in editing the GRUB configuration file, you may see the following message the next time you boot Linux:

```
   Booting 'Red Hat Enterprise Linux ES (2.4.21-9.EL)'

root (hd0,0)
 Filesystem type is ext2fs, partition type 0x83
kernel /vmlinuz-2.4.21-9.EL ro root=LABEL=/

Error 15: File not found

Press any key to continue
```

This is just one of many possible boot problems. Sometimes the boot disk floppy you created during the installation process can help. If you've misplaced this disk, the `mkbootdisk` command can help.

However, the boot disk may not help you in every case. And if you don't have a boot disk, Red Hat Enterprise Linux has other automated recovery options. You can use any standard installation boot disk, even the first installation CD, to rescue your Linux system. Depending on the problem, you could select the automated recovery process or start Linux in single-user mode.

TIP *Another option for a rescue CD is the Knoppix distribution; it allows you to load a complete Linux operating system from CD, with many of the same tools described in this book for repairing Linux. It has a wider array of tools than is loaded from a Red Hat rescue disk. For more information, see* `www.knoppix.net`.

The Specialized Boot Disk

The easiest way to get around the specified problem is with a boot disk. The boot disk that you should have created during Red Hat Enterprise Linux installation is customized for this purpose. As long as you haven't changed the way partitions are organized, the custom boot disk should start your Linux system.

It's easy to create a new boot disk with the `mkbootdisk` *`versionnumber`* command, where you use the version number associated with your Linux kernel. For example, if the kernel shown in the `/boot` directory is `vmlinuz-2.4.21-13.EL`, the following command creates a customized boot disk on a 1.44MB floppy:

```
# mkbootdisk 2.4.21-13.EL
Insert a disk in /dev/fd0. Any information on the disk will
➥ be lost. Press <Enter> to continue or ^C to abort:_
```

Just remember to test your customized boot disk as soon as possible. You don't want problems with this disk when you're trying to rescue your Linux system.

Rescue Mode

Customized boot disks don't solve all possible Linux boot problems. Fortunately, you're not out of luck. Even if you've lost your customized boot disk, Red Hat's `linux rescue` mode will normally get you into your Linux system. Once you've started Linux, you can restore or repair any damaged files that you have.

TIP *To use* `linux rescue` *mode, you need access to the Red Hat Enterprise Linux installation files. If you're starting from a network boot disk, you need the address and location of the* `/RedHat` *directory. See Chapter 4 for examples.*

You can start `linux rescue` mode from any Red Hat Enterprise Linux installation boot disk or CD. If you don't have one available, you can download it from `ftp.redhat.com` or associated mirror sites. You can even create installation boot disks on a Microsoft Windows computer using the `RAWRITE.EXE` utility discussed in Chapter 3. Just type `linux rescue` at the `boot:` prompt for installing Red Hat Enterprise Linux, like so:

```
boot: linux rescue
```

At this point, you may wonder if you did the right thing, because Red Hat Enterprise Linux takes you through the first two steps of a standard installation: language and keyboard type. If you used a

Red Hat Enterprise Linux installation floppy or boot CD, you'll also need to enter the location (local or network) of the Red Hat Enterprise Linux installation files, as if you were installing from a network. Refer to Chapter 4 if you need more information.

If you're starting from the first Red Hat Enterprise Linux installation CD, you'll be asked whether you need to set up networking at this point. It's not required, because installation files are available on the first installation CD. Now `linux rescue` mode presents a menu with three different options, as shown in Figure 11.7:

Figure 11.7
Options in linux rescue mode

Continue If you select Continue, Red Hat Enterprise Linux searches your hard disk for your installation. All located filesystems are mounted as subdirectories of `/mnt/sysimage`. I think of this as automatic rescue mode.

Read-Only The Read-Only option is almost identical, except that located filesystems are mounted in read-only mode. You can think of this as read-only rescue mode.

Skip The Skip option proceeds directly to a root shell prompt in single-user mode. No attempt is made to look through available filesystems. I view this as a manual rescue mode.

Once you've made the necessary changes, type the `exit` command. Repeat as needed until you see messages regarding termination signals. Linux should unmount all filesystems and then automatically reboot your computer.

Automatic Rescue Mode

If automatic rescue mode is successful, Red Hat Enterprise Linux mounts all appropriate filesystems from `/etc/fstab` on `/mnt/sysimage`. In this case, the `df` command reflects the mounted directories, as shown in Figure 11.8.

FIGURE 11.8 Rescue mode mounts

```
sh-2.05b# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      2619      3151  46% /
/dev/root.old             6120      2619      3151  46% /
/tmp/cdrom              462464    462464         0 100% /mnt/source
/dev/hda2              3644000   3031676    427976  88% /mnt/sysimage
/dev/hda1               101089     15035     80835  16% /mnt/sysimage/boot
/dev/hdd1              1031800    568924    410464  59% /mnt/sysimage/home
sh-2.05b#
```

In Figure 11.8, the CD is mounted on `/mnt/source`, `/dev/hda2` is mounted on `/mnt/sysimage`, `/dev/hda1` is mounted on `/mnt/sysimage/boot`, and `/dev/hdd1` is mounted on `/mnt/sysimage/home`. While it's easy to see that `/dev/hda1` is associated with `/boot` and `/dev/hda2` is associated with root (`/`), you can confirm this with the following `e2label` *`partitiondevice`* command:

```
# e2label /dev/hda1
/boot
```

But what if automatic rescue mode can't mount all of your filesystems? In this case, you may see an error message such as the following:

```
Error mounting filesystem on hdd1: Invalid argument
```

Simply continue with automatic rescue mode. Linux mounts as many filesystems as it can. In this case, you can work on any damage to an unmounted filesystem such as `/dev/hdd1`.

If you have one or more unmounted filesystems, the first two things to check are the `fstab` configuration file and the integrity of the format itself. At this point, you can use the `vi` editor to check `fstab`, but since the root (`/`) directory is actually mounted on `/mnt/sysimage`, you'll need the following command to open `fstab`:

```
# vi /mnt/sysimage/etc/fstab
```

Alternatively, to clean up a damaged, unmounted filesystem, use the `fsck` *`devicename`* command. For example, to check `/dev/hdd1`, run the following command:

```
# fsck /dev/hdd1
```

TIP *If you want to access the Linux man pages in* `linux rescue` *mode, run the* `chroot /mnt/sysimage` *command. This restores your top-level root (*`/`*) directory to the top of the hierarchy, activating the standard paths to the Linux man pages.*

Read-Only Rescue Mode

The only difference between read-only and automatic rescue mode is that all filesystems are mounted in read-only mode. This may be the best choice if you have a large number of filesystems, such as with a typical server installation of Red Hat Enterprise Linux.

You can remount any desired filesystem in read-write mode. For example, the following command remounts partition device `/dev/sda2` on the root (`/`) directory in read-write mode:

```
# mount -w -o remount /dev/sda2 /
```

NOTE *This command is equivalent to the* `mount -o remount,rw /` *command described in Chapter 7.*

Manual Rescue Mode

Sometimes `linux rescue` mode can't find any of your filesystems. Don't panic; the problem could be as simple as an error in the name of `/etc/fstab`. Manual rescue mode is the most appropriate here.

This mode loads a minimal root image and the kernel to a RAM disk, and then it sends you to a root shell prompt (`#`). No filesystems are mounted; you have access only to a basic set of commands, such as `mount`, `mkdir`, `mv`, `cp`, `rm`, `fdisk`, and `fsck`. Once you've mounted a directory, you can also use the `vi` editor to change the files you need.

But remember, this is a minimalist version of Linux. You don't have all the commands that you may be used to at this level.

In manual rescue mode, the first step is to mount the partition associated with your root (`/`) directory in a temporary location such as `/mnt/sysimage`. This should allow you access to additional commands from directories such as `/bin`, `/sbin`, and `/usr/sbin`.

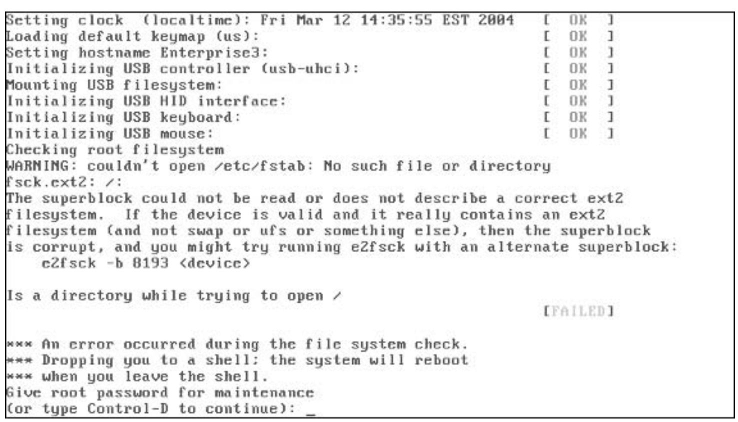
Single-User Mode

There's one other method, known as *single-user mode*, that you can use to log into a damaged Linux system. If Linux can find your root (`/`) directory filesystem, it can start Linux in this mode. As described earlier, single-user mode, also known as runlevel 1, requires only two services.

Once you've made any required changes, you don't have to reboot. The `exit` command automatically moves you to the default runlevel as defined in `/etc/inittab`. Alternatively, the `init 3` or `init 5` commands can immediately start those respective runlevels. Single-user mode is also useful for changing the root password. If you forgot the password, run the `passwd` command in single-user mode. The password you enter becomes the new root password.

Sometimes you'll encounter a problem such as a bad `/etc/fstab` file or an unmountable filesystem during the boot process. In this case, you'll see a prompt similar to that shown in Figure 11.9. When you enter the root password at the prompt, Linux starts in single-user mode.

FIGURE 11.9
Dropping to single-user mode



```
Setting clock  (localtime): Fri Mar 12 14:35:55 EST 2004   [  OK  ]
Loading default keymap (us):                               [  OK  ]
Setting hostname Enterprise3:                              [  OK  ]
Initializing USB controller (usb-uhci):                    [  OK  ]
Mounting USB filesystem:                                   [  OK  ]
Initializing USB HID interface:                            [  OK  ]
Initializing USB keyboard:                                 [  OK  ]
Initializing USB mouse:                                    [  OK  ]
Checking root filesystem
WARNING: couldn't open /etc/fstab: No such file or directory
fsck.ext2: /:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

Is a directory while trying to open /
                                                           [FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue): _
```

You can also start single-user mode from the GRUB menu. As described earlier, it's easy to protect GRUB with a password. If you don't see GRUB editing options as shown in Figure 11.10, enter the `p` command and then enter the GRUB password.

FIGURE 11.10 GRUB editing commands

Highlight the Linux operating system of your choice and then press the `a` command to modify the kernel arguments. GRUB should take you to a line such as the following:

```
grub append> ro root=LABEL=/ hdc=ide-scsi
```

At this point, you can add a command to the end of this line, such as `single`, `1`, or `init=/bin/sh`, as shown in Figure 11.11.

FIGURE 11.11 Modifying GRUB for single-user mode

When you press Enter, Red Hat Enterprise Linux proceeds to boot in single-user mode, runlevel 1. At this point, you can `fsck` unmounted drives, edit configuration files, check the status of LVM partitions, and more.

Other Runlevels

Single-user mode isn't the only useful troubleshooting option. You can boot into the runlevel of your choice through the GRUB boot menu.

Many regular Linux workstations are configured to boot into the GUI through runlevel 5. When successful, it brings up one of the display managers shown in Chapter 29. But there are several potential problems with starting the GUI, including the following:

- A nonfunctioning X Font Server
- A full `/tmp` directory partition
- A full `/home` directory partition

These problems would also make it difficult to open another text login console. You may even need to reset this computer with the power button. However, that doesn't prevent you from booting Linux into a different runlevel such as 3, which brings you to a text login console. You can then check the noted problems using commands described elsewhere in the book.

Summary

In this chapter, you learned about the Linux boot process. The basic process starts with the computer BIOS. Once it detects basic hardware on your system, it points to the Linux bootloader (GRUB), where you can select an operating system. When you select Red Hat Enterprise Linux, the bootloader starts the kernel. The `/etc/inittab` file then starts the processes associated with the default runlevel.

It helps to have the customized boot disk that you created during Linux installation or with the `mkbootdisk` command. It can help you start Linux even when you have a number of different problems in the boot process. However, `linux rescue` mode, using one of the Red Hat Enterprise Linux installation boot disks, is also a viable option. In various `linux rescue` modes, you can `fsck` partitions, edit configuration files, and more. Alternatively, you can start Linux in single-user mode or other runlevels, which can help you address other problems, such as a lost root password or a nonfunctional X interface.

NOTE *In Red Hat parlance, it's common to refer to a partition check by its command; in other words, you can* `fsck` *(pronounced* fisk*) a partition.*

In the next chapter, you will learn about the Linux kernel in detail. Once you understand the basics, it is not difficult to modify, recompile, and implement a new Linux kernel.

Chapter 12

Upgrading and Recompiling Kernels

THE THOUGHT OF RECOMPILING a kernel strikes fear into many Linux users. It is true; errors in this process can lead to an unbootable system. If you don't have an appropriate backup, recovery can be difficult. But with a few simple precautions, you can avoid risks when you recompile a kernel. Once you understand the basic steps, it is not a difficult process.

There is an easy way to upgrade a kernel: just install the next version of the Red Hat kernel RPM that's customized for your CPU. The Red Hat Enterprise kernel RPM automatically updates your bootloader so you can start Linux with either the old or the new kernel.

The Red Hat kernel RPM may not include the very latest upgrades. The latest Linux kernels are available in tarball format; alternatively, minor upgrades require only a patch. This chapter describes both options. But the Red Hat Enterprise kernel includes a number of "backports" from kernel version 2.6, so it may already meet your needs.

You can customize and recompile the kernel already on your computer, or you can download, customize, and recompile a new kernel. The wide variety of options makes this process seem more difficult than it really is. In this chapter, you'll learn about three different `make` kernel configuration tools.

This chapter includes a detailed analysis of what you can change, based on the GUI kernel configuration tool. This tool is organized into configuration menus, storage devices, networking, other hardware support, and other software support categories. We've included a step-by-step summary at the end of the chapter. Many of you will want to read the summary first to get a quick sense of what you'll need to do.

I've included a number of different kernel version numbers. The Linux kernel version released with Red Hat Enterprise Linux 3 is 2.4.21. But remember, as Red Hat has included backports from kernel 2.6, it's actually more advanced than the generic Linux 2.4.21 kernel. Some version numbers in this chapter may be higher, which can reflect the changes that you or a colleague may already have made.

Once you've made the desired changes, you need to compile your new kernel. It's a straightforward, step-by-step process. After compiling the kernel, you'll want to copy it to the appropriate directories.

At least for now, you'll also want to configure it into your bootloader as though the old and new kernels were two different operating systems. This chapter covers the following topics:

- Why bother?
- "Upgrading" the easy way
- Exploring sources, tarball, and patch alternatives
- Customizing a kernel
- Setting up configuration menus
- Kernels, section by section
- Updating the bootloader

Why Bother?

The kernel that comes with Red Hat Enterprise Linux works for most hardware and software applications. But you may want to change your kernel for any of the following reasons:

Drivers You want to take advantage of a new driver. It may be for hardware you just installed or for a filesystem you want to try.

Bugs You've learned that your current Linux kernel doesn't work in some way that affects how you run this operating system.

Features You've heard about a new kernel. Perhaps it provides improved hardware support, such as for an IEEE 1394 FireWire video recorder. Maybe it allows you to connect to a backup jukebox of 1,000 writeable DVDs.

Security You may want to protect yourself against a newly discovered security breach.

Size It's possible to speed up your system by removing unneeded drivers, thereby reducing the size of your kernel.

When you want to change your kernel, you should consider the following options, in order:

1. Recompile your existing kernel. New kernels and associated source code can consume a lot of space. A newer kernel may not work as well with the software you have in place. You may be able to do what you need with the existing kernel.
2. Upgrade your current kernel. Tools such as `up2date` and `yum` that we described in Chapter 10 can even help you automate this process.
3. Patch your existing kernel. You can perform small upgrades of Linux kernels with a patch. When applied, the patch is incorporated into your current kernel source code. For example, a single patch can upgrade your kernel from version 2.4.21 to 2.4.22. However, this may cause more trouble than its worth for a Red Hat Enterprise kernel.
4. Install a new kernel. Once the new kernel package is installed, you should also configure and compile the new kernel.

Kernel Version Numbers

Linux kernels are stored in the /boot directory with a name such as vmlinuz-2.4.23. All kernels include a version number in a *major.minor.patch* numbering format. In this case, the first number (2) refers to the second major release of the Linux kernel. The second number (4) has two meanings: it's the fourth minor release of the specified major kernel, and since it's an even number, it's a production-ready version of the kernel. The third number (23) refers to the twenty-third patch to the specified minor release.

Red Hat and Fedora Linux kernels have version numbers that look slightly different, such as 2.4.23-10. You can see that there's an extra number; this is the build number. Each "build" can incorporate a small number of new drivers or bug fixes. Red Hat Enterprise Linux kernels include one more extension, EL. The kernel originally released with this operating system is 2.4.21-4.EL, which includes some of the "backports" from Linux kernel 2.6 that we described in Chapter 1. Some experimental kernels include a number with a "pp," a "pre-patch," which is a test release of a kernel. Other variations are available, such as nptl, which is one extension on a Fedora kernel with Native POSIX Threaded Library support (already included in Red Hat Enterprise Linux).

If you're installing a new kernel on a production computer, avoid odd minor numbers; for example, kernel version 2.5.22 is a beta release not suitable for the real world. In addition, pre-patch (pp) kernel releases may also be fraught with risk.

"Upgrading" the Easy Way

Red Hat makes it easy to "upgrade" a kernel. If you're willing to use the "stock" Red Hat packaged kernel RPM, you can install the next version of your kernel with little trouble.

Furthermore, if you install a Red Hat kernel RPM, the new kernel is added to your bootloader as if it were a different operating system. If you have problems with the new kernel, all you need to do is reboot and select the older kernel in your bootloader.

Installing the Newest Red Hat Kernel

While you may be used to upgrading RPMs, it's best to install (instead of upgrading to) the latest kernel RPM. Yes, that means you'll have two Linux kernels installed, side by side. One example is shown in Figure 12.1. RPMs on this network are mounted on the /mnt/inst directory.

Upgrades are riskier because it's more difficult to go back. If you delete your kernel, your system may stop working completely. You may be able to go into linux rescue mode, described in Chapter 11, to remove the upgraded kernel and then reinstall the original; but then again, you may need to reinstall Linux.

FIGURE 12.1
Installing a new kernel RPM

```
[root@Enterprise3 root]# rpm -ivh /mnt/inst/RedHat/Updates/kernel-2.4.21-9.EL.i6
86.rpm
Preparing...                ########################################### [100%]
   1:kernel                 ########################################### [100%]
[root@Enterprise3 root]# 
```

Several Red Hat `kernel-*` RPMs are available, and they can be customized by CPU. Red Hat Enterprise Linux kernel RPM files are organized in the following format:

```
kernel-versionnumber.cputype.rpm
```

Red Hat customizes kernels for the CPU types shown in Table 12.1. Red Hat may not provide the latest Linux kernel in RPM format customized for your CPU. To find your ***`cputype`***, use the following command:

```
# uname -p
```

NOTE *For the rest of this chapter, I'll substitute* `x` *for* `versionnumber` *in files and directories.*

TABLE 12.1: CUSTOM RED HAT KERNELS

CPU TYPE	DESCRIPTION
alpha	From the HP alpha CPU, developed by the former Digital Equipment Corporation
athlon	For the AMD Athlon CPU
i586	Intel 586 CPU
i686	Intel 686 CPU
ia64	Intel Itanium 64-bit CPU
ppc	Power PC CPU
ppc64	Power PC, 64-bit CPU
s390	Specialty CPU for an IBM server
s390x	A 64-bit version of the s390

NOTE *Keep good records of the RPMs you've installed. Start with* `/root/install.log`*, which is a list of RPMs installed when you installed Red Hat Enterprise Linux on your computer.*

It's easy to set up a new kernel from its RPM. Once you have access to the RPM, you just need to install it, similar to the command shown in Figure 12.1. Alternatively, if the kernel RPM filename is `kernel-2.4.22-3.EL.i686.rpm`, located in the `/mnt/inst` directory, just run the following command:

```
# rpm -ivh /mnt/inst/kernel-2.4.22-3.EL.i686.rpm
```

If you see a `Failed dependencies` error, install packages listed in your error message first. The actual packages you may need to install or upgrade will depend on the requirements of the new kernel and what you already have installed.

When you install another kernel, you're installing several files in the `/boot` directory. These files are stored side by side with files associated with your old kernel. We describe these files in Table 12.2.

And that's it! Your new kernel is automatically installed. Wasn't that easy? In the next section, you'll see what the newly installed kernel does to your bootloader.

TABLE 12.2: KERNEL-RELATED */BOOT* FILES

FILE	DESCRIPTION
`config-*`	Kernel configuration file; you can read the settings.
`initrd-*`	Initial RAM disk; allows the kernel access to drivers at the start of the boot process.
`module-info-*`	A list of available hardware modules for this kernel.
`System-map-*`	A memory map with different functions.
`vmlinux-*`	The kernel.
`vmlinuz-*`	A compressed version of the kernel.

TIP *If you accidentally upgrade to (and not install) the latest kernel, it'll overwrite the original kernel settings. You can recover. Just force the reinstallation of the original kernel with the* `rpm -ivh --force kernel-`***`x.cputype`***`.rpm` *command.*

Bootloader Updates

The Red Hat Enterprise Linux kernel RPMs automatically update your active bootloader, whether it be GRUB or LILO. Detailed information on each bootloader is available in Chapter 11. A revised `grub.conf` file with two different kernels is shown in Figure 12.2.

This particular `grub.conf` file makes it look as if you have a choice between the following three different operating systems:

- Red Hat Enterprise Linux (new kernel number)
- Red Hat Enterprise Linux (old kernel number)
- DOS (typically, a version of Microsoft Windows)

FIGURE 12.2
An updated GRUB bootloader

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,1)
#          kernel /vmlinuz-version ro root=/dev/hda3
#          initrd /initrd-version.img
#boot=/dev/hda
password --md5 $1$Uq7cGO$8Ro6Hj7ESm7A5QUyOAyGIO
default=1
timeout=10
splashimage=(hd0,1)/grub/splash.xpm.gz
title Red Hat Enterprise Linux ES (2.4.21-9.EL)
        root (hd0,1)
        kernel /vmlinuz-2.4.21-9.EL ro root=LABEL=/ hdc=ide-scsi
        initrd /initrd-2.4.21-9.EL.img
title Red Hat Enterprise Linux ES (2.4.21-4.EL)
        root (hd0,1)
        kernel /vmlinuz-2.4.21-4.EL ro root=LABEL=/ hdc=ide-scsi
        initrd /initrd-2.4.21-4.EL.img
title DOS
        rootnoverify (hd0,0)
        chainloader +1
"grub.conf" 24L, 825C
```

Remember, the kernel is the core of the operating system. Thus, when you install a new kernel, you've actually installed another version of Linux. Yet both kernels still use most of the same utilities, programs, and commands.

You may also note the `default=1` command, which actually points to the second stanza as the default operating system. In other words, if you don't select a different operating system in 10 seconds (`timeout`), GRUB automatically boots your old Red Hat Enterprise Linux kernel.

You can see the result in Figure 12.3, which shows the associated GRUB menu. Note that the second listing for Red Hat Enterprise Linux, with the original kernel number, is highlighted.

FIGURE 12.3
The revised GRUB menu

NOTE *The default Red Hat Enterprise Linux bootloader is GRUB. The Red Hat installation program saves a version of LILO in* `/etc/lilo.conf.anaconda`*. If you make a copy of this file in* `/etc/lilo.conf`*, the Red Hat kernel RPM will automatically upgrade LILO as well.*

Kernel Version 2.6

Red Hat includes a number of features of the latest Linux kernel version 2.6 in its enterprise kernels. In fact, if you "upgrade" to a generic kernel 2.6, you may lose some features. Red Hat has "back-ported," or incorporated, a number of kernel 2.6 features in its EL kernels, including the following:

- Native Posix Thread Library support, which supports multithreaded applications. Linux historically hasn't distinguished between threads and processes in the Microsoft fashion.
- IPSec support, for faster secure network transmissions through the kernel.
- Asynchronous Input/Ouput, which means that applications don't have to wait for reads and writes to files.
- O(1) Scheduler, also known as "The Big O," which is key to the scalability of Red Hat Enterprise Linux. For the uninitiated, the key features are summarized in the Linux Gazette at `www.linuxgazette.com/issue89/vinayak2.html`.

- OProfile support, which allows CPU-based performance monitoring at low overhead. For more information, see `oprofile.sourceforge.net/about`.
- kksymoops support, which improves bug tracking and reporting.
- Reverse Map Virtual Memory, which uses linked lists to track memory pages; this improves performance especially when RAM is limited.
- HugeTLBFS, which supports larger memory applications, as well as memory hotplugging.
- Remap_file_pages allows the kernel to rearrange and optimize paged memory.
- Network stack features of kernel 2.6, which improves support for IPv6 addressing and the current Internet Group Management Protocol (IGMPv3).
- IP virtual server (IPvs) support for network load balancing.
- Access Control Lists (ACLs) for more finely grained file security.
- 4GB/4GB memory split allows up to 8GB of physical memory on x86 computers.
- Scheduler support for hyperthreaded CPUs allows a single CPU to act as multiple virtual processors.

NOTE *This list of "backports" includes "oversimplifications"; those who are really familiar with the Linux kernel may find the list less than complete. On the other hand, this list may seem like gobbledygook to most Linux users. I just want to highlight the kernel features that Red Hat has included in Red Hat Enterprise Linux 3, so everyone can use them as needed.*

Exploring Sources, Tarballs, and Patch Alternatives

One drawback to using Red Hat kernel RPMs is that they may not incorporate the latest features into the latest kernel. If you need the absolute latest kernel, you're probably going to have to download and process a tarball package. More information on the `tar` command is available in Chapter 14; we'll go through the process step by step here. However, you should first try to work with the Red Hat Enterprise kernel, especially since it's supported by Red Hat.

The Red Hat Enterprise Kernel Source

Remember, if you install a kernel not developed for Red Hat Enterprise Linux 3, you may be giving up features that Red Hat has backported from earlier versions of kernel 2.6. The alternative is to install and customize the Red Hat kernel; you'll need to install the `kernel-source` RPM associated with your kernel version.

For example, the `kernel-source` RPM, for 32-bit Intel systems, associated with the original release of Red Hat Enterprise Linux 3, is `kernel-source-2.4.21-4.EL.i386.rpm`. If you're using one of the update CDs, you can find updated `kernel-source` RPMs in the `RedHat/Updates` directory.

In most cases, the Red Hat Enterprise source code should be good enough. Unlike the stock kernels described later, the code is supported by Red Hat. You should first try the source code for your architecture.

For example, if you've upgraded to the `2.4.21-9.EL` kernel, you can install the source code from the first Red Hat Enterprise update CD. Assuming it's mounted in the `/mnt/cdrom` directory and you have a computer with an Intel 32-bit architecture, you can do so with the following command:

```
# rpm -Uvh /mnt/cdrom/RedHat/Updates/kernel-source-2.4.21-9.EL.i386.rpm
```

If this is good enough for you, feel free to skip to the "Customizing a Kernel" section.

Download Sources

The Linux kernel is under constant development. As new features emerge, a loose team of volunteers headed informally by Linus Torvalds decides when new kernels are ready for test and production release. You can download their work from the `kernel.org` Internet sites. If you can't get to `www.kernel.org` or `ftp.kernel.org`, you can select from mirror websites all over the world, as listed in `www.kernel.org/mirrors`.

I've demonstrated a download of Linux kernel 2.6.4 from `ftp.kernel.org` in Figure 12.4. This particular package is large; at around 43MB, it might take a few hours to download on a regular telephone modem.

Setup

Now that you've downloaded the kernel tarball package, the setup process is easy. The package you've downloaded should have a name similar to `linux-x.tar.gz`. With that in mind, follow these steps:

1. Copy the kernel tarball package to the `/usr/src` directory. For example, if the package is in the current directory, run the following command:

   ```
   # cp linux-x.tar.gz /usr/src
   ```

2. Navigate to `/usr/src`, and unpack the tarball. The following commands should open a large number of files in the `/usr/src/linux-x` directory:

   ```
   # cd /usr/src
   # tar xzvf linux-x.tar.gz
   ```

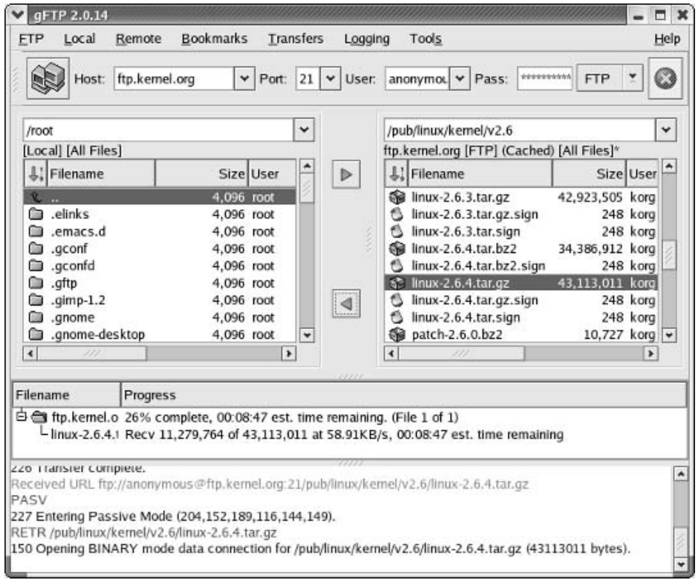
3. Navigate to the `/usr/src/linux-x` directory. Later in this chapter, you'll learn how to open and use a Linux kernel configuration menu. You'll then compile and install your new kernel.

The Patch Alternative

Installing and compiling a new kernel may seem like a lot of trouble, just to upgrade for the latest changes. Assume your computer has Linux kernel 2.6.3 installed. If you just want to incorporate the new security features available in Linux kernel 2.6.4, you have an alternative: you can install a kernel patch.

Warning *Don't upgrade a Red Hat Enterprise Linux kernel with a stock patch. While the version numbers are in the 2.4 series, you could lose features that Red Hat has incorporated from the 2.6 series kernels. In addition, the standard patch process described in this section could run into other errors.*

FIGURE 12.4
Downloading a kernel tarball



You can download kernel patches from the `kernel.org` Internet or mirror sites described earlier. As of this writing, they're available from the same directories as regular kernels, with a simple name: `patch-x.gz`, where `x` represents the upgraded version number.

NOTE *We assume that you've already installed the source code for the current kernel via the* `kernel-source-*` *RPM. The other required RPM packages are described later in this chapter, in Table 12.3.*

Once you've downloaded the patch, copy it to the `/usr/src` directory. Then you can upgrade your kernel with the following command:

```
# zcat patch-x.gz | patch -p0
```

This command reads from the compressed patch file, identifies the differences with the current Linux kernel's source code, and updates files as needed. If there are problems, they're documented in `*.rej` files in your kernel source directory, `/usr/src/linux-2.4`.

You can then customize, compile, and install your patched kernel as described later in this chapter.

Customizing a Kernel

Customizing the kernel is a long process. In this section, you'll look at the basic steps. Later in the chapter, you'll go through the graphical configuration menu in detail. The following are the basic steps that we detail in the rest of this chapter:

1. You should have already downloaded the source code for the kernel, from the `kernel-source` RPM or some generic kernel file.

2. You'll download the development tools that help you customize the kernel, as we describe in the "Setting Up Configuration Menus" section. Then you'll navigate to the source code directory, normally linked to `/usr/src/linux-2.4`.
3. Edit the `Makefile` to label your new kernel. Clean up your source files with the `make mrproper` command.
4. Set up the `.config` configuration file for your kernel.
5. Revise your kernel settings; we'll show you the menus you can use to help.
6. Work through the dependencies with the `make dep` command.
7. Make sure the source code files are clean with the `make clean` command.
8. Create the new kernel with the `make bzImage` command.
9. Assuming you've configured modules, create the associated directories with the `make modules` and `make modules_install` commands (in that order).
10. Install the kernel, set up an initial RAM disk, `config` file and more in the boot directory with the `make install` command.

If you want to save your kernel configuration in the `/boot` directory, you'll need to copy the `.config` file to that directory, with an appropriate name such as `config-2.4.21-4.ELbluesman`.

Kernels vary. The steps you need to take may vary. The first thing you should do is read the README file in the `/usr/src/linux-2.4` directory for any major changes in procedure.

With hundreds of different settings, I can't cover all the problems that you may see. Pay attention to any error messages, especially those that result from the various `make` commands. When I see a problem, I use Google (`www.google.com`) and Google Groups (`groups.google.com`) to search for others who have encountered similar problems. Sometimes simple solutions, such as using the `make menuconfig` instead of `make xconfig`, can help.

Note *If you're using the Red Hat kernel source code RPM, the* `/usr/src/linux-2.4` *directory is automatically linked to the actual directory with the source code.*

Preparing the Source

You may have downloaded a new kernel. Perhaps you just want to change some settings on the kernel that you're currently using. In either case, you need to prepare the source code.

Remember, the source code is located in the `/usr/src/linux-x` directory, where `x` stands for the `linux-versionnumber` of your kernel. Navigate to this directory. Read the following sections in sequence. We'll assume you've installed the Red Hat `kernel-source` RPM, where the `/usr/src/linux-2.4` directory is linked directly to the source code that we'll be compiling. The steps are slightly different if you're compiling a stock kernel downloaded from a non–Red Hat source.

Tip *It's important to follow these instructions in the right order when revising and recompiling your Linux kernel. If you have problems, refer to the Linux Kernel* `HOWTO` *at* `www.tldp.org`.

The *Makefile*

Open the file named `Makefile` from the `/usr/src/linux-2.4` directory in your favorite text editor. The first four lines in this file should look similar to the following:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 21
EXTRAVERSION = something
```

If you're new to Linux kernels, this may be confusing. The labels in the `Makefile` are not consistent with the standard kernel numbering format: `PATCHLEVEL` is the minor version revision level of the kernel, and `SUBLEVEL` is the patch revision level of the kernel.

`EXTRAVERSION` is what Linux adds to the end of the kernel files that you can transfer to the `/boot` directory at the end of this process. It also helps you identify your new kernel in a bootloader such as GRUB.

Change the `EXTRAVERSION` variable to something you'll recognize. For the purpose of this chapter, I'm editing my `Makefile` with the following:

```
EXTRAVERSION = -9.ELbluesman
```

Tip *Be very careful with the* `EXTRAVERSION` *variable; an extra space after* `bluesman` *would create an error during the kernel module configuration process.*

Saving the Current Kernel Configuration

If you've never revised your kernel before, Red Hat already has your current kernel configuration on file. As described earlier, it's located in the `/boot/config-x` file. This is also true if you've installed a different Red Hat Enterprise Linux kernel from a "stock" kernel RPM.

If you've recompiled your kernel before, your current configuration should be in the hidden file, `.config`, in the `/usr/src/linux-2.4` directory. Save it now. The step described in the next section deletes that file.

In either case, back up your current configuration. These files are small enough to fit on a regular floppy disk.

Cleaning the Source

Now that your `Makefile` is ready, it's time to clean the source code. If you aren't already there, navigate to the `/usr/src/linux-2.4` directory. The following command uses the `Makefile` script to clean files and directories that would interfere with compiling the kernel source code:

```
# make mrproper
```

Tip *Each of the* `make` *commands in this chapter may run through thousands of lines of code. While some may take minutes, others may take hours, especially on slower computers. Be patient.*

A Standard Starting Point

When you download a kernel from a non–Red Hat source such as `ftp.kernel.org`, you may have to adjust several hundred settings to match the current Red Hat configuration. That process can be painful.

Alternatively, you can set a Red Hat starting point for your kernel; some may call this a *baseline configuration*. The following are four basic options for your baseline Linux kernel; each is mutually exclusive:

The saved `.config` file If you saved the `.config` file earlier, you can restore it to the `/usr/src/linux-2.4` directory.

The `/boot/config-x` file This file contains the configuration of your kernel when you installed it from an RPM or when you installed Red Hat Enterprise Linux. You can copy this to the `/usr/src/linux-2.4/.config` file.

Your current configuration Use the `make oldconfig` command to set up your current configuration in the `/usr/src/linux-2.4/.config` file.

The appropriate file in `/usr/src/linux-2.4/configs` The `configs` subdirectory includes a series of configuration files, customized for different CPUs. You can copy the file closest to your kernel to the `/usr/src/linux-2.4/.config` file.

Customizing the Configuration

You've seen three menus you can use to customize your kernel configuration: `make config`, `make menuconfig`, and `make xconfig`. Select one and make the desired changes to your kernel, using the techniques and criteria described earlier. Generally, you'll want to do the following:

- Use modules. Make sure they're enabled in the Loadable Module Support menu. The alternative is to use a monolithic kernel, which may be too big for your system.
- Be sure to cite the correct CPU in the Processor Type And Features menu.
- Remove unneeded devices and modules. This can minimize the size of your kernel and associated driver files. For example, if you're not planning to connect a Ham Radio to your Linux computer, you won't need the modules associated with Amateur Radio Support.
- If in doubt, don't remove it. Assuming you're starting from a baseline or standard kernel configuration, many of the settings are interdependent. If you remove the wrong device, you can make this kernel unusable.

NOTE *Previous kernels required symmetric multiprocessing (SMP) support, even for computers with one CPU. This is no longer required for the kernel included with Red Hat Enterprise Linux 3.*

When you've made your changes, save your configuration. By default, the `make` tools save your settings to the `.config` file in the `/usr/src/linux-2.4` directory.

While you could edit the `.config` file directly, Linux includes a number of special menus that can help you work through all of the settings. This is a long process, which we describe later in this chapter.

Creating Dependencies

Now you can force your source code to read your Linux kernel configuration. The following command resolves all dependencies. It takes the settings from your new `.config` file and uses them to customize your source code.

```
# make dep
```

NOTE *The* `make dep` *process took about 10 minutes on my 2.4GHz computer. Your experience depends in part on the speed of your CPU and the size of your kernel.*

Next, you'll want to clean the source in preparation for the following steps. This command removes unused files from your new configuration:

```
# make clean
```

Making a Kernel Image

Now that the dependencies are satisfied, you're ready to "make" the kernel image. This process can take minutes or even all night, depending on the speed of your CPU. You want the image to be compressed so that you can fit it on a boot or rescue floppy disk. To create a compressed kernel image, run the following command:

```
# make bzImage
```

You'll see a long series of messages. When the `make bzImage` command is complete, without errors, watch for the following message:

```
warning: kernel is too big for standalone boot from floppy
```

If this is what you see, you probably can't use the `mkbootdisk` command from Chapter 11 to create a boot floppy. If you're motivated to make your kernel smaller, you may want to start the `make xconfig` process again and remove more settings.

NOTE *The* `make bzImage` *process took 20 minutes on my 2.4GHz computer. Your experience depends in part on the speed of your CPU and the size of your kernel.*

You may not need a customized boot disk. In many cases, you can use the Red Hat Enterprise Linux installation boot disk in rescue mode to boot your system. For more information on the `linux rescue` process, see Chapter 11.

TIP *The Red Hat Enterprise Linux installation boot disk in* `linux rescue` *mode may not rescue all systems. You may need a customized boot disk for your new kernel.*

With previous versions of Red Hat and Red Hat Enterprise Linux, you had to manually copy the kernel from the directory cited in the last message. It's no longer required. After you create the appropriate module directories in the next two sections, you'll run the `make install` command to move the kernel and update the boot loader automatically.

Building Modules

At this point, we've assumed you've configured module support into your kernel. The next step is to "make" your modules. The first command organizes the modules you've configured in various `/usr/src/linux-2.4` subdirectories.

```
# make modules
```

Note *The* `make modules` *process took 70 minutes on my 2.4GHz computer. As always, your experience depends in part on the speed of your CPU and the size of your kernel.*

The next command organizes your modules in the `/lib/modules/2.4.21-9.EL` directory.

```
# make modules_install
```

Note *The* `make modules_install` *process took just a few minutes on my 2.4GHz computer.*

Tip *If you see the* `when making multiple links, last argument must be a directory` *error message, check the* `EXTRAVERSION` *variable in the* `/usr/src/linux-2.4/Makefile`*. There may be an extra space at the end of that line.*

Remember, this was just an overview. Now you're ready to get to the nitty-gritty of customizing the kernel, based on one of three different configuration menus. But before you start, rerun the steps described in the "Preparing the Source" and "Customizing the Configuration" sections.

I've summarized the steps at the end of the chapter. In that summary, you'll rerun the first four steps before starting a configuration menu.

Setting Up Configuration Menus

If you're going to customize your kernel in any way, you need a configuration menu. Different menus are available in text, terminal graphics, and GUI formats. Each of these menus requires a series of packages: the source code and language libraries for the kernel and the language libraries for graphical configuration screens.

Once you've set up the menu of your choice, it's just a tool in the kernel configuration process. Use it in the steps summarized at the end of this chapter.

Kernel RPM Packages

Several RPMs associated with building a kernel are available, as shown in Table 12.3. Some provide the source code; others are related to languages and libraries needed to configure and process the kernel. Install them using the `rpm` command described in Chapter 10.

Alternatively, you can open the Package Management menu in the Linux GUI and install the Development Tools and Kernel Development package groups. We described how you use this tool in Chapter 10. This is one case where it may be faster to use a Red Hat GUI tool, instead of the command-line interface. However, this process installs more software than you need just to open the menus used to configure the Linux kernel.

TABLE 12.3: KERNEL RPM PACKAGES

PACKAGE	DESCRIPTION
`binutils-*`	Required binary utilities
`cpp-*`	A GNU C language preprocessor
`gcc-*`	The C language compiler
`glibc-devel-*`	For developing programs (such as the kernel) that require C language libraries
`glibc-kernheaders-*`	Kernel C language header files
`kernel-source-*`	Kernel source files
`ncurses-*`	A language library for presenting graphics on a terminal; required for `make menuconfig`
`ncurses-devel-*`	Header files for `ncurses` screens
`tcl-*`	TCL scripting language; designed for use with TK; required for `make xconfig`
`tk-*`	Widgets for GUIs designed to work with TCL; required for `make xconfig`

TIP *If you get a* `Failed dependencies` *message related to* `kernel-headers`*, install the* `glibc-kernheaders-*` *RPM package. Many dependencies explicitly cite the RPM package that you need. Dependencies related to* `kernel-headers` *do not.*

If you're reconfiguring an existing kernel, you don't need to install the `kernel-x.cputype.rpm` package. You'll actually be creating a new kernel from some of the other packages when you compile it later in this chapter.

If you're willing to customize your kernel in text mode, you don't need the `ncurses*` or `tcl-*` or `tk-*` RPM packages. But a kernel contains a huge number of settings that you can customize, which makes the graphical kernel configuration screens a terrific convenience. You'll see this for yourself in the following section.

Make Menus

Now that you have the right RPM packages installed, it's time to examine the three different menus available for customizing your kernel. Start by navigating to the directory with your Linux kernel's source files, `/usr/src/linux-versionnumber`. For convenience, we'll refer to the linked `/usr/src/linux-2.4` directory for the rest of this chapter.

TIP *By default, the Red Hat Enterprise Linux* `kernel-source` *RPM links the* `/usr/src/linux-2.4` *directory to the default source code directory for your original kernel.*

You'll find a `Makefile` in `/usr/src/linux-2.4` that lets you configure your kernel. That file includes the following three kernel configuration tools:

- `make config`

- `make menuconfig`
- `make xconfig`

We introduce these tools briefly in the following sections. Then we'll use `make xconfig` to analyze what you can configure in your kernel in detail.

Before moving on, navigate to the `/usr/src/linux-2.4` directory on your computer. The `make` commands shown won't work unless you're in that directory.

Why a Menu?

You can edit your configuration file directly. As described earlier, your first kernel configuration is documented in the `config-x` file (where *x* is the version of your kernel), in your `/boot` directory. This file includes all kinds of settings, such as the following:

```
CONFIG_MODULES=y
CONFIG_3C359=m
# CONFIG_IRDA_DEBUG is not set
```

In other words, the `CONFIG_MODULES` setting, which lets your kernel use modular drivers, is integrated into the kernel. The `CONFIG_3C359=m` command turns this particular network card driver into a module; when Red Hat detects this card, it will be able to use the `insmod` command (see Chapter 11) to use this driver. Unused elements such as `CONFIG_IRDA_DEBUG` are left out of the kernel and modules; the hash mark (#) turns it into a comment, and your kernel ignores the line.

When you're done, you should save the file to `.config` in the `/usr/src/linux-2.4` directory. Then you'll be ready to compile and install your kernel, as described later in this chapter.

Tip *If you've recompiled your kernel before, the settings are normally saved in the* `/usr/src/linux-2.4/.config` *file. One previous revision is saved in* `/usr/src/linux-2.4/.config.old`*. Nevertheless, this is a good time to back up your* `.config` *file to another directory.*

But because this file contains about 2,000 lines, analyzing each line can be a time-consuming process. For that reason, three kernel tools are available to help.

make config

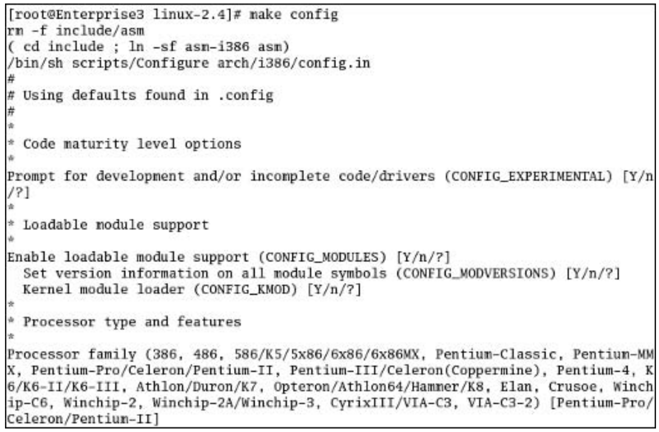
When you're in the `/usr/src/linux-2.4` directory, the `make config` command starts a kernel configuration tool. It prompts you with a series of questions, as shown in Figure 12.5.

It starts by looking for a `.config` file in your `/usr/src/linux-2.4` directory. If that file does not exist, it uses `uname -p` to identify your CPU and find the corresponding file in the `/usr/src/linux-2.4/configs` directory. The settings in the selected file become your default values.

Alternatively, if you're modifying your kernel for the first time, you can use the installed configuration in `/boot/config-x`, where *x* represents the kernel version number. Copy it to `/usr/src/linux-2.4/.config` with the following command:

```
# cp /boot/config-x /usr/src/linux-2.4/.config
```

FIGURE 12.5
The *make config* process



```
[root@Enterprise3 linux-2.4]# make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in .config
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n
/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
  Set version information on all module symbols (CONFIG_MODVERSIONS) [Y/n/?]
  Kernel module loader (CONFIG_KMOD) [Y/n/?]
*
* Processor type and features
*
Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic, Pentium-MM
X, Pentium-Pro/Celeron/Pentium-II, Pentium-III/Celeron(Coppermine), Pentium-4, K
6/K6-II/K6-III, Athlon/Duron/K7, Opteron/Athlon64/Hammer/K8, Elan, Crusoe, Winch
ip-C6, Winchip-2, Winchip-2A/Winchip-3, CyrixIII/VIA-C3, VIA-C3-2) [Pentium-Pro/
Celeron/Pentium-II]
```

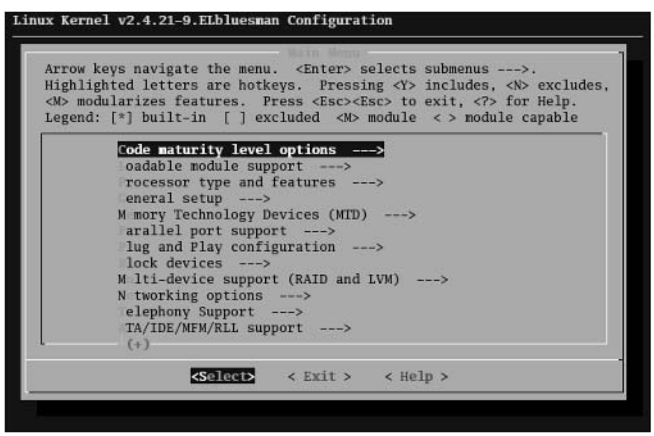
Next, you get a bunch of questions. For each question, you have up to four options. `Y` and `N` are straightforward. In many cases, you can select `M`, which makes the relevant driver module available in a file. And if you enter `?`, you open a help file related to the question.

But you need to answer hundreds of questions. If you just have to change the setting for the 366th question, you may miss it. If you pass a question, there is no way to go back. You just have to press Ctrl+C and start the process again. For this reason, the other two "make" menu options are more popular.

MAKE MENUCONFIG

When you're in the `/usr/src/linux-2.4` directory, the `make menuconfig` command should give you a low-resolution graphical menu. As long as you have the `ncurses*` RPM packages installed, as described earlier, you should see a menu similar to Figure 12.6.

FIGURE 12.6
The make *menuconfig* main menu



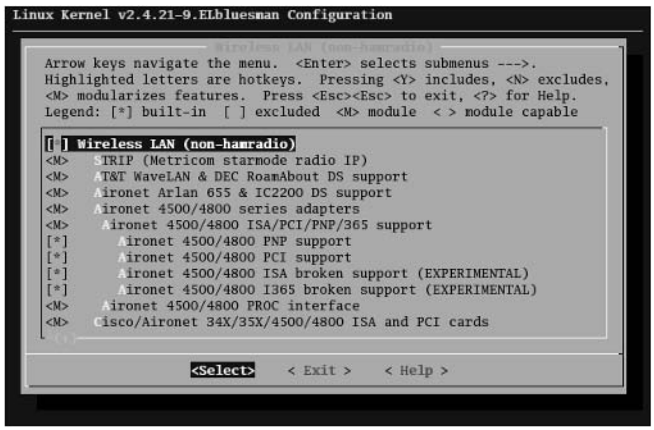
NOTE *Look at the title of the menu. You'll see that it includes the custom settings from your* `Makefile` *in the* `/usr/src/linux-2.4` *directory.*

Like `make config`, this option looks for a `.config` file in the `/usr/src/linux-2.4` directory. If it does not exist, it uses the `*.config` file customized for your CPU in the `/usr/src/linux-2.4/configs` directory.

As you can see, kernel settings are organized into menus. You can highlight a setting and select Help at any time. Unfortunately, help is not available for every variable.

Highlight a menu option, and press Enter to review detailed configuration options; for example, Figure 12.7 illustrates some available wireless LAN devices. As you can see, some are available modules, and others are built into the kernel.

FIGURE 12.7
The Wireless LAN kernel menu



NOTE *You can run* `make menuconfig` *on a Telnet or SSH connection from a remote computer. Depending on your point of view, this may be a convenience or a security risk. More information on Telnet and SSH is available in Chapter 18.*

As you can see, many other menus are available through `make menuconfig`. We illustrate these options in detail in the next section, since the `make xconfig` menus are easier to read in a book.

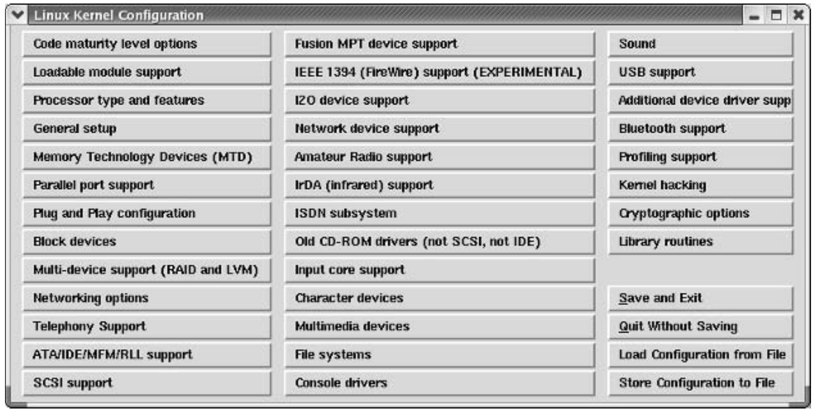
When you exit from `make menuconfig`, you get a chance to save your new configuration. If you select Yes, this tool writes your new kernel configuration to `/usr/src/linux-2.4/.config`.

MAKE XCONFIG

When you're in the `/usr/src/linux-2.4` directory and a GUI, the `make xconfig` command should give you a high-resolution graphical menu. As long as you have the `tcl-*` and `tk-*` RPM packages installed, you should see a menu similar to Figure 12.8.

Like `make config`, this option looks for a `.config` file in the `/usr/src/linux-2.4` directory. If it does not exist, it uses the `*.config` file customized for your CPU in the `/usr/src/linux-2.4/configs` directory. If you want to start with a different configuration file, the two buttons in the lower-right corner can help.

FIGURE 12.8
The Linux Kernel Configuration menu



As you can see, different kernel settings are organized into different sections. We'll look at these sections in much more detail later in this chapter. When you're happy with your changes, click Save And Exit; otherwise, click Quit Without Saving.

Kernels, Section by Section

What follows is a section-by-section analysis of the Linux kernel, based on the `make xconfig` Linux Kernel Configuration menu. This is a fairly long section, so if you're reading this full chapter, you may want to take a break.

If you're using this section to configure your kernel, don't forget to run the first basic steps; we've summarized them as the first four steps at the end of this chapter.

A total of 34 kernel menus are shown; I've organized them into six sections. While I've tried to follow the `xconfig` menu as closely as possible, they are not in the order shown in Figure 12.8.

- Basic configuration menus help you configure the fundamental parts of the kernel, such as the CPU and ISA or PCI support. Be especially careful with these menus; errors can keep Linux from recognizing peripherals or even your CPU.
- Storage device menus help you work with connections related to all types of storage: hard drives, CDs, parallel port drives, and more. Be careful; you want to make sure Linux can recognize your hard disks.
- Networking menus allow you to configure basic network software and network hardware in detail.
- External hardware covers menus associated with hardware that's physically outside the computer box.
- Other hardware support is associated with hardware that does not easily fit into any of the other categories.
- Other software support includes critical components such as filesystems and libraries.

If you want to follow along on your Linux computer, navigate to the `/usr/src/linux-2.4` directory and run the `make xconfig` command. As you go through each section, click on the applicable button in the Linux Kernel Configuration menu.

Examine the hardware kernel settings with a critical eye. If you know that you'll never use the associated hardware, consider deactivating the setting. If you might add the noted hardware in the future, consider creating a module. These actions minimize the size of your kernel and can greatly improve the startup speed and performance of your system.

WARNING *If in doubt about an active or modular kernel setting, don't deactivate it. There are a number of innocuous-looking kernel parameters that are critical to the basic operation of Linux.*

Basic Configuration Menus

I've arbitrarily organized several menus in this section. They include the basic parameters associated with starting Linux, recognizing hardware, setting up a CPU, and using experimental components in the kernel.

NOTE *Previous versions of the kernel included a Binary Emulation Of Other Systems menu, which allowed users to configure support to emulate other Unix-style systems, including UnixWare 7.x, SCO Open Server, and Solaris 2.x.* This option is no longer available in the kernel included with Red Hat *Enterprise* Linux 3.

Code Maturity Level Options

If you're using Red Hat Enterprise Linux in an organization that prohibits "experimental" Linux drivers, make sure the setting shown in Figure 12.9 is set to `n`. Otherwise, you may accidentally include experimental kernel drivers and settings. However, many officially "experimental" Linux drivers work well today, including IEEE 1394 (FireWire) support; `CONFIG_EXPERIEMENTAL` is enabled by default.

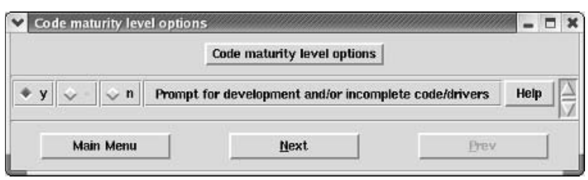


FIGURE 12.9
Code Maturity Level Options menu

If you're a developer, be careful. It's a good idea to work on only one experimental driver at a time; if you have problems, you'll know the source. You can find more information on each variable by clicking the associated Help button. The help dialog box for this menu is shown in Figure 12.10.

Loadable Module Support

Normally, when Red Hat Enterprise Linux detects new hardware on your computer, it automatically installs the driver module, if available. This is possible in part to the Loadable Module Support options shown in Figure 12.11.

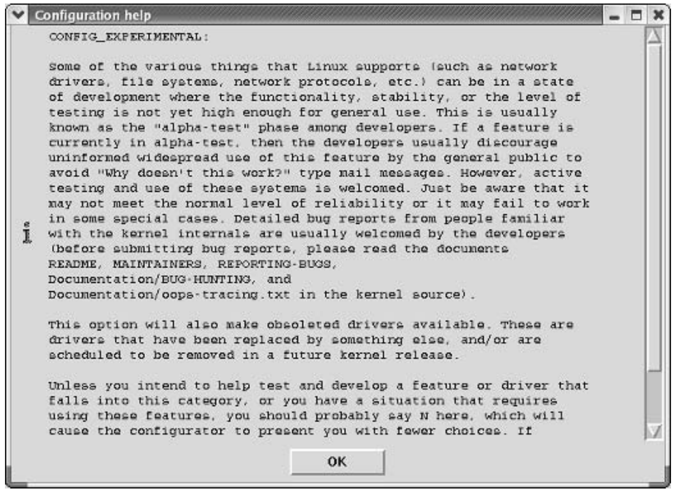


FIGURE 12.10
Configuration Help menu

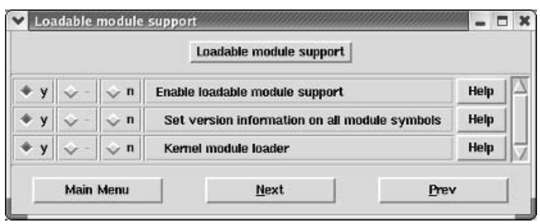


FIGURE 12.11
Loadable Module Support menu

You should almost always answer yes to all of these options; they allow you to separate hardware driver modules from the kernel, use drivers from different sources, and load modules as needed.

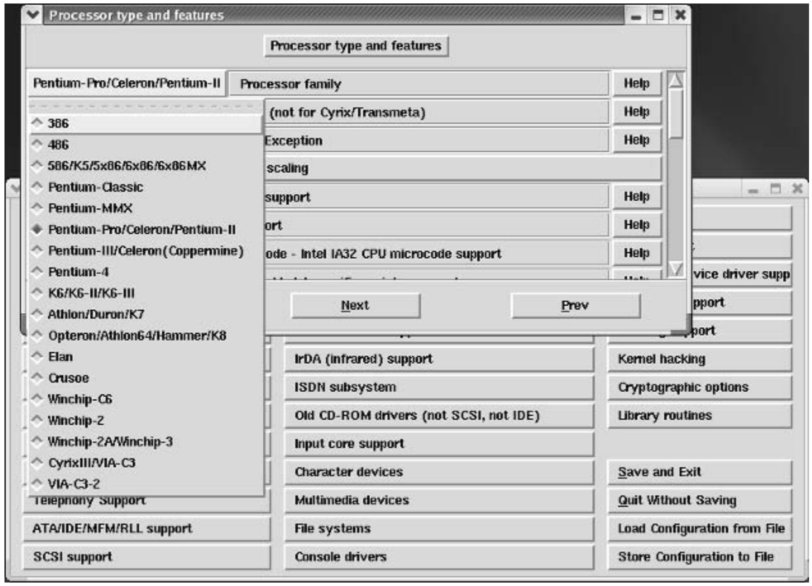
Otherwise, you'd have to include all possible drivers in the main kernel. This would make the kernel large and unwieldy. Some kernels without modules, also known as *monolithic kernels*, are so big that older PCs aren't able to load them when you try to boot Linux.

Processor Type and Features

You can customize the Linux kernel for your CPU. This loosely corresponds to the different kinds of `kernel-x.cputype.rpm` packages that you can install directly on your computer. As you can see in Figure 12.12, you can configure the kernel for a wide variety of CPUs.

If you don't see your CPU in the list, find the closest available match. If you have an Intel 32-bit CPU, you can also try 386 for a basic kernel good for all current Intel 32-bit CPUs. Naturally, if you have a different CPU such as an Itanium, you'll be working with a different kernel menu, and the options will be different.

Figure 12.12
Processor Type And Features options



Once you've selected the processor, you should configure a number of other variables, including special modules that can support multiple CPUs and special features of Toshiba or Dell laptops.

General Setup

The General Setup kernel menu shown in Figure 12.13 provides several basic hardware, binary, and networking options for the kernel. Look through the list of variables. They fall into a number of categories and some are fundamental to the kind of hardware on your computer. These categories include the following:

Networking support Some programs require kernel networking support even if your computer never connects to another network or the Internet.

Basic hardware support Normally, Linux kernels are configured with support for PCI, ISA, and PCMCIA cards.

Hot-pluggable support Linux can be configured to support hardware that can be installed or removed while your computer is running.

Power management support Linux supports the older Advanced Power Management (APM) system; while support for the Advanced Configuration Power Interface (ACPI) standard is still experimental, it works fine on my laptop computer. I've just added the `apm=off acpi=on` commands to the `kernel` line in my `grub.conf` bootloader configuration file.

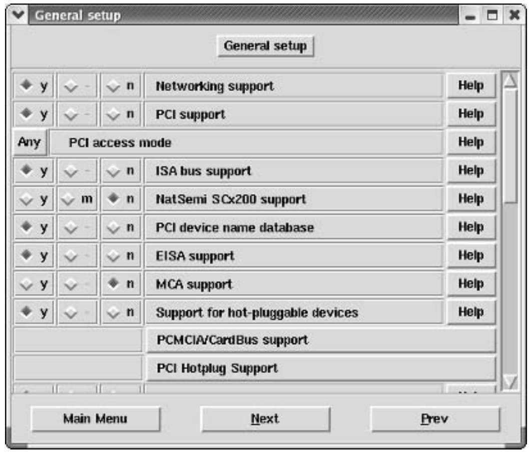


FIGURE 12.13
General Setup menu

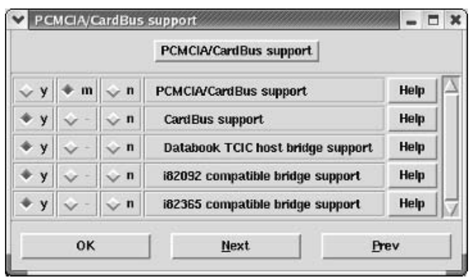


FIGURE 12.14
PCMCIA/CardBus Support menu

NOTE *In the General Setup menu, click the PCMCIA/CardBus Support button. You'll see the submenu shown in Figure 12.14. If you're using Linux on a laptop computer, be sure that the appropriate bridges are active.*

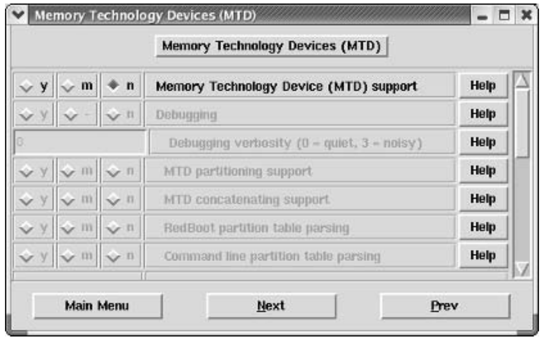
Storage Devices

Several Linux kernel menus organize the settings related to where you can store files and other information. If you have an external storage device, see the menus described in the "External Hardware" section.

MEMORY TECHNOLOGY DEVICES

In the Linux kernel, Memory Technology Devices (MTD) includes everything that can store information in a "solid state." Examples include the BIOS, camera flash cards, and ROM chips. Remember, some of these can be installed through a PCMCIA adapter. The basic menu is shown in Figure 12.15; these devices are disabled by default in Red Hat Enterprise Linux 3.

FIGURE 12.15
Memory Technology Devices (MTD) menu

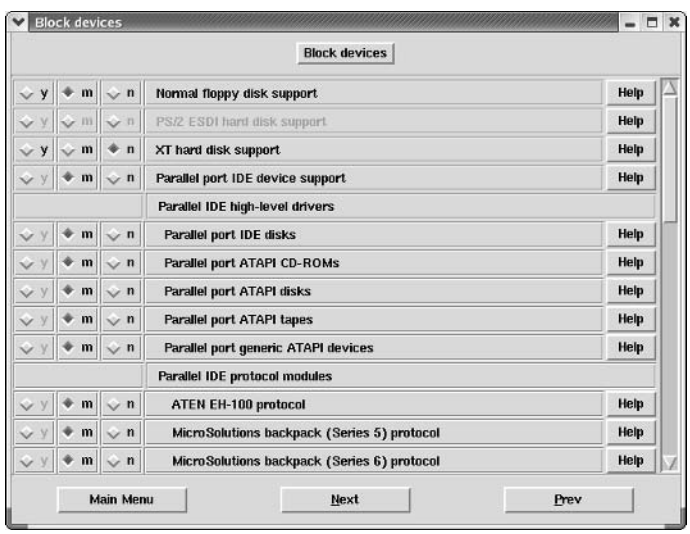


BLOCK DEVICES

Block devices allow you to mount a storage unit, such as a floppy or a hard drive, on a directory. Open the Block Devices menu, and you'll see something similar to Figure 12.16. Scroll down the menu. You'll see support for floppy drives, regular IDE hard disks, shared network drives, and RAM disks. Some special drivers are available, such as for the older MicroSolutions external "backpack" hard drive that connected through a parallel port. USB and FireWire block device drivers have their own categories covered later in this chapter.

These settings are closely related to ones found on the ATA/IDE/MFM/RLL Support menu.

FIGURE 12.16
The Block Devices menu



Multi-Device Support

As described in Chapter 3, Red Hat Enterprise Linux supports RAID and LVM. Both systems require multiple partitions. Since Linux assigns a device to each partition, RAID and LVM are considered multidevice systems. If you ever intend to use RAID or LVM, you should activate these settings, as shown in Figure 12.17.They are modular by default, which means the drivers are installed whenever you configure RAID and or LVM.

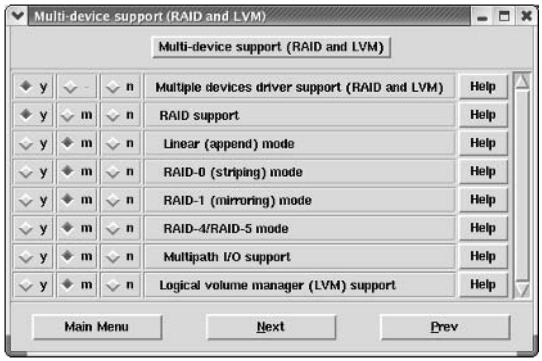


Figure 12.17
Multi-Device Support (RAID and LVM) options

ATA/IDE/MFM/RLL Support

ATA, IDE, MFM, and RLL are a bunch of acronyms all related to standard PC hard disk and CD-ROM interfaces. As shown in Figure 12.18, there's an IDE, ATA And ATAPI Block Devices button that you can click to call up a submenu with variables for different drives and chipsets.

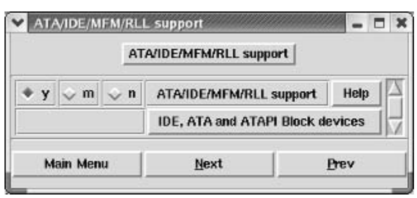


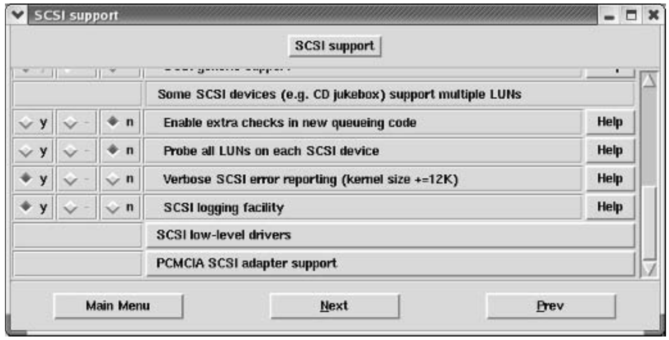
Figure 12.18
ATA/IDE/MFM/RLL Support menu

SCSI Support

The other major interface for storage devices is SCSI, the Small Computer Systems Interface. The SCSI Support kernel menu allows you to activate drivers or modules for basic SCSI hard drives, tape drives, and CD systems. At the bottom of the SCSI Support menu shown in Figure 12.19, there are two submenus:

- The SCSI Low-Level Drivers menu includes support for a number of specific SCSI hard drives and RAID devices.
- The PCMCIA SCSI Adapter Support menu accommodates PCMCIA cards that connect your computer to SCSI devices.

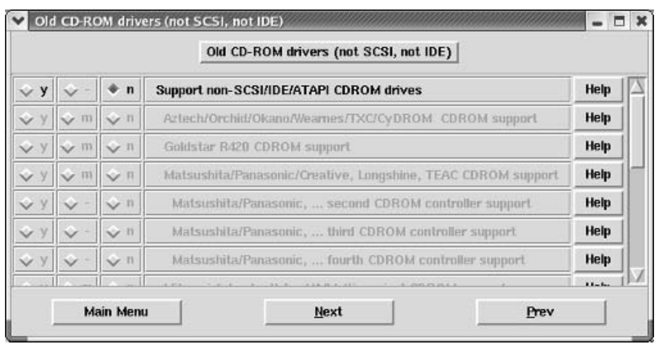
FIGURE 12.19
SCSI Support options



OLDER CD-ROM DRIVERS

Older CD-ROM drives were connected to sound cards. The Old CD-ROM Drivers (Not SCSI, Not IDE) menu includes access to the Linux drivers that were once used for these drives. A number of drivers are available, as shown in Figure 12.20.

FIGURE 12.20
The Old CD-ROM Drivers (Not SCSI, Not IDE) menu



As you can see, older CD-ROM drivers are organized by make and model. If you have an older CD-ROM drive that's not on this list, check your documentation. Try the driver associated with a similar make or model. Just remember, these drivers are no longer supported and may not work well with the latest Linux production kernels. They're not active by default for Red Hat Enterprise Linux 3.

Networking

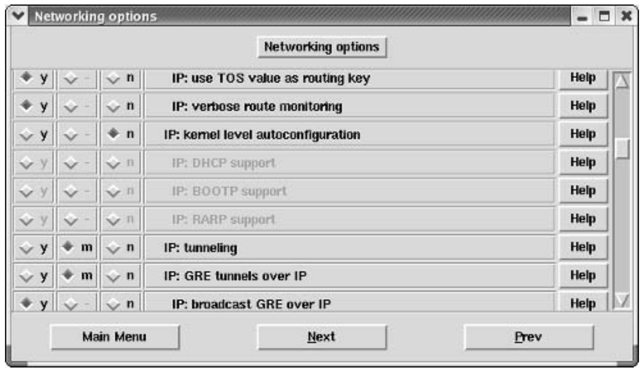
Linux is built for networking. Naturally, it offers several networking-related kernel configuration menus. You can configure basic network software as well as specific devices in the various Networking menus. Not all of these menus are strictly related to networking.

More information on basic network protocols is available in Chapter 15. Other important reference chapters for Linux kernel network settings are Chapters 16 and 17.

Networking Options

The Networking Options menu is primarily used to configure network software. Although you can activate other protocol stacks such as IPX/SPX, many of the options relate to the primary network protocol for Linux and the Internet, TCP/IP. This is a large menu; part of it is shown in Figure 12.21.

Figure 12.21
Networking Options menu



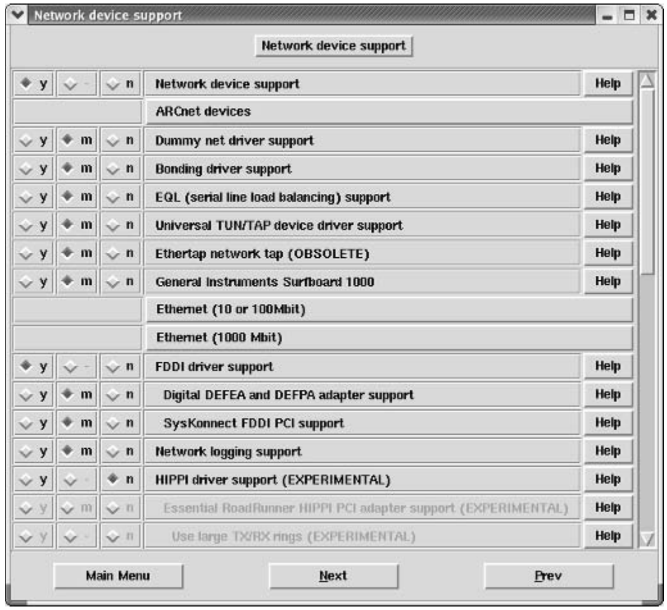
Many of these settings aren't obvious; for example, the IP: DHCP Support option shown in Figure 12.21 is used only for remote Linux terminals. Remember, the help menus provide more information on each setting. Several submenus are available, as follows:

- The IP: Netfilter Configuration submenu allows your kernel to support firewalls using `iptables`, `ipchains`, or even `ipfwadm`. The `ipchains` commands are associated with Linux kernel 2.2; the `ipfwadm` commands are obsolete and are associated with Linux kernel 2.0. You can learn more about the current `iptables` firewalls in Chapter 17.
- The IP Virtual Server configuration menu includes features from kernel 2.6 that support load balancing of your network.
- The IPv6: Netfilter Configuration submenu allows you to configure firewalls if you're using this more advanced system of IP addressing, described briefly in Chapter 15. Remember, IPv4 is still in common use today.
- The Appletalk Devices menu allows you to communicate with Apple computers over a TCP/IP network.
- The QoS And/Or Fair Queuing menu supports networks that allow you to prioritize messages, using "Quality of Service" parameters.
- The Network Testing menu lets you send preconfigured data packets to check the capacity of your system.

Networking Devices

The Network Device Support kernel menu allows you to activate any number of drivers for different kinds of network adapters. This is also a substantial menu, as shown in Figure 12.22.

FIGURE 12.22
Network Device Support menu



It includes a list of basic network drivers and several submenus with hardware-specific drivers. As you can see, network cards were developed for a number of different network systems, such as Ethernet. These submenus include the following:

- ARCnet Devices allows you to use network cards built for a specific type of LAN. ARCnet is a variation on Token Ring; because it's a slow network (2.5Mbps), it is generally not used today.
- Ethernet (10 Or 100 Mbit) lets you configure regular and Fast Ethernet adapters. If you don't see your adapter in this list, check your documentation for "clones." For example, many older network cards can use the Novell NE2000 driver.
- Ethernet (1000 Mbit) permits you to configure Gigabit Ethernet network adapters on your Linux computer.
- Wireless LAN (Non-Hamradio) allows you to configure basic wireless networking on your PC, mostly for devices that conform to the IEEE 802.11b standard. Bluetooth support is available under a separate menu. As of this writing, third-party drivers for IEEE 802.11a and 802.11g devices are available through Linuxant (`www.linuxant.com`).
- The Token Ring Devices submenu lets you configure specific network adapters designed for this older network system. While Token Ring networks are not in common use, some believe that they are more reliable than Ethernet; thus, you may still find some of these networks in places such as factories.

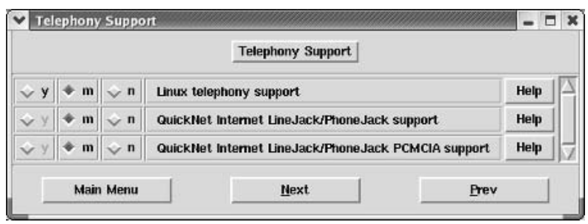
- Wan Interfaces permits you to configure network devices that connect two distant LANs in a Wide Area Network (WAN).
- PCMCIA Network Device Support allows you to accommodate network cards to this standard, primarily for laptop computers.
- The ATM Drivers submenu let you adapt network cards built for Asynchronous Transfer Mode (ATM) networks. ATM is a popular alternative to Fast and Gigabit Ethernet.

Tip *If you're configuring your kernel for a network card that conforms to the PCMCIA or PC Card standard, check the PCMCIA Network Device Support menu.*

Telephony Support

Modern telephone companies translate regular phone calls to data that's often sent over networks such as the Internet. This process is known as *telephony*. Linux supports a couple of telephony cards, primarily used to help larger businesses translate phone calls to data. The Telephony Support menu is shown in Figure 12.23.

Figure 12.23
Telephony Support menu

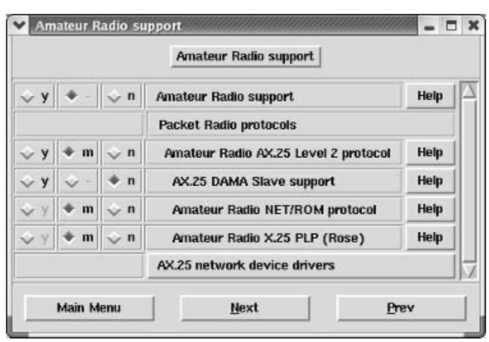


Amateur Radio

You can configure the Linux kernel to support connections to amateur radios, as shown in the Amateur Radio Support menu in Figure 12.24.

Computers can be networked through amateur radios, using the AX.25 protocol. There is even an AX.25 Network Device Drivers submenu that allows you to configure this type of network connection.

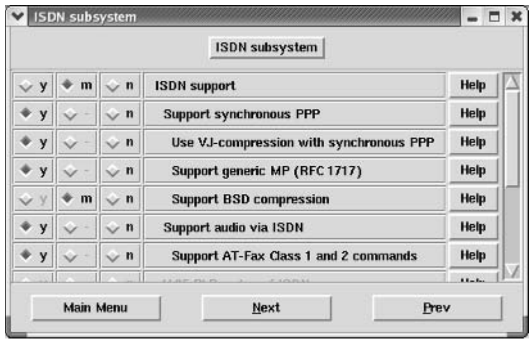
Figure 12.24
Amateur Radio Support menu



ISDN

Early digital computer connections over telephone networks were made using Integrated Services Digital Network (ISDN) adapters. These connections are still popular in Europe, and are often the only "high-speed" wired (128Kbps) option in rural areas of the United States of America. The basic ISDN Subsystem menu shown in Figure 12.25 allows you to configure ISDN with several types of networks and commands.

FIGURE 12.25
ISDN Subsystem menu



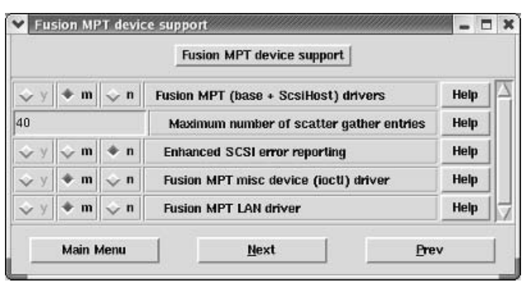
The Linux kernel ISDN Subsystem menu includes of the following submenus:

- The ISDN Feature Submodules submenu allows you to configure a virtual ISDN card and some commands that may be needed for European connections.
- The Passive ISDN Cards submenu lets you configure adapters that are generally used by consumers; they're associated with 128Kbps speeds.
- The Active ISDN Cards submenu allows you to configure higher-speed ISDN adapters.

Fusion MPT

Fusion MPT Device Support is a specialty menu for high-speed SCSI devices from LSI Logic. There is also an associated LAN driver, as shown in Figure 12.26.

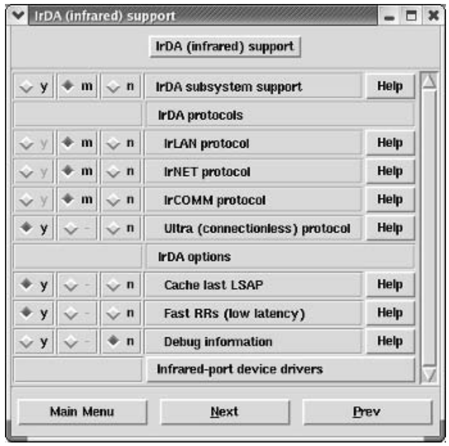
FIGURE 12.26
Fusion MPT Device Support menu



INFRARED

You can configure the Linux kernel to work with infrared devices that conform to the Infrared Data Association (IrDA) standard. As you can see from the menu in Figure 12.27, there are several infrared protocols for transmitting data. The Infrared-Port Device Drivers submenu allows you to include the appropriate hardware in the Linux kernel or modules.

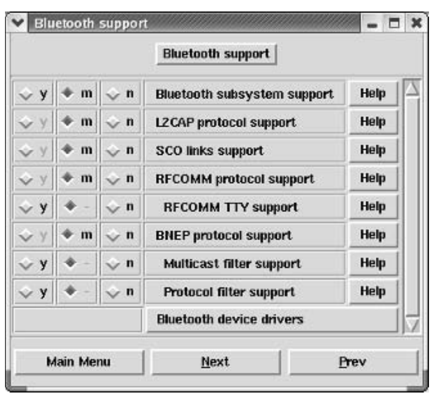
FIGURE 12.27
IrDA (Infrared) Support menu



BLUETOOTH

The Bluetooth specification is based on a radio technology for networks. The range is short—typically around 33 feet (10 meters). It's commonly used on portable devices such as handheld computers and cellular telephones. Several portable devices are built on Linux. Bluetooth technology can also be used to connect regular computers in networks. The kernel Bluetooth Support menu is shown in Figure 12.28.

FIGURE 12.28
Bluetooth Support menu



The Bluetooth Device Drivers submenu allows you to use the basic Host Controller Interface (HCI). Different drivers are available for USB, serial ports, and the PCMCIA cards associated with various vendors.

External Hardware

You'll see three Linux kernel menus for external hardware. Two are related to relatively new standards: USB and IEEE 1394. The other menu addresses older external hardware: parallel port support.

Parallel Port Support

The parallel port is commonly known as the *printer port*. As you can see in Figure 12.29, you can configure parallel port support in several ways. For example, IEEE 1284 transfer modes support standard bidirectional communication with a printer.

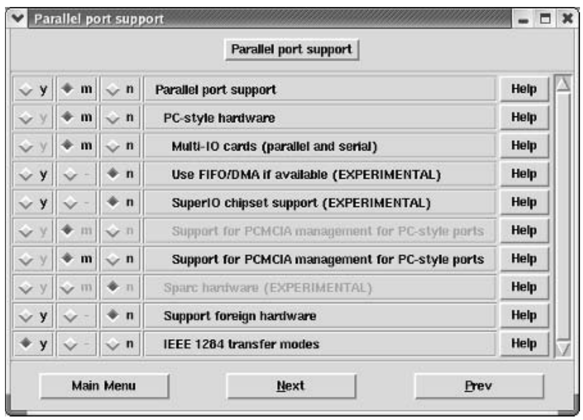


FIGURE 12.29 Parallel Port Support menu

Remember, parallel ports aren't just for printers. For example, you can connect a number of hard disks and other storage devices to the parallel port. It's also a way to sync computers and transfer data. More information is available under the Block Devices menu.

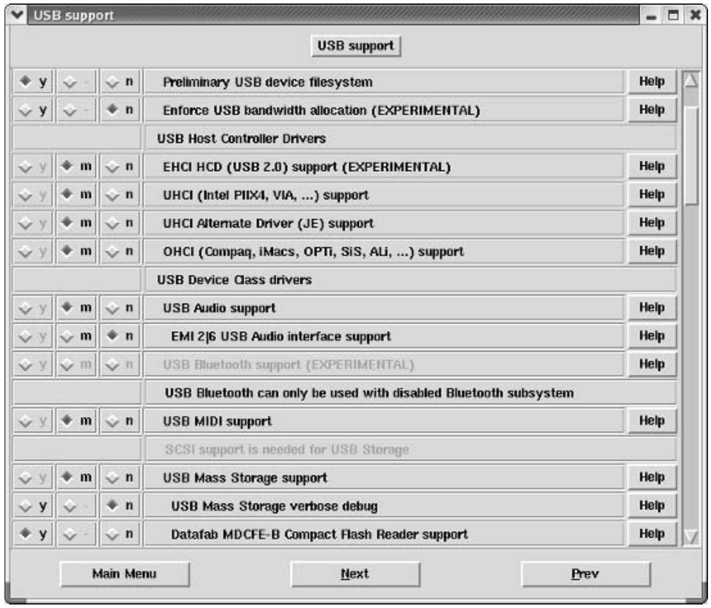
USB Support

It seems possible that all future external devices will conform to some USB or IEEE 1394 standard. New hardware in both areas is being released at a fast and furious pace. Linux developers are working steadily to keep up.

Linux support for USB is far from complete; kernel support for USB 2.0 standard devices is still *officially* experimental as of this writing. More information on Linux and USB is available in Chapter 2 and from `www.linux-usb.org`. As you can see from the main USB Support menu shown in Figure 12.30, kernel code is available for the major types of USB hardware.

The USB Serial Converter Support submenu allows you to configure serial port adapters. This lets you connect a serial device, such as an older mouse, to an USB port.

FIGURE 12.30
USB Support menu



IEEE 1394: FireWire/iLink

As discussed in Chapter 2, IEEE 1394 hardware is more popularly known by its trade names, FireWire and iLink. Linux support for these devices is still experimental. Associated devices use its high-speed (400Mbps+) capabilities, as shown in Figure 12.31.

NOTE *This menu is not active if you've deactivated the development drivers setting in the Code Maturity Level Options menu.*

WARNING *Experimental code is not production-ready. In other words, testing is not complete, and the associated kernel components may not work and could even affect other parts of your system. However, many experimental devices work well; for example, I've even connected a FireWire external hard drive to my Linux computer. Use hardware associated with experimental code at your own risk.*

Other Hardware Support

Some hardware menus are difficult to put in any of the other categories. Several are related to the ways terminals and consoles work locally and remotely; there's also plug and play, and there's multimedia.

Character Devices

A *character device* transfers data to and from a user process and is often associated with a serial port. The most common character device is a terminal. You can configure drivers for local virtual terminals as well as remote terminals. Some remote terminals can use serial and other physical ports. You can review these options in Figure 12.32.

FIGURE 12.31
IEEE 1394 (FireWire) Support (EXPERIMENTAL)

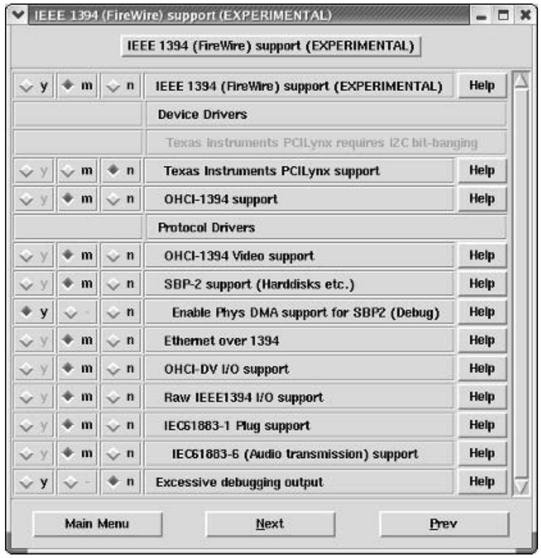
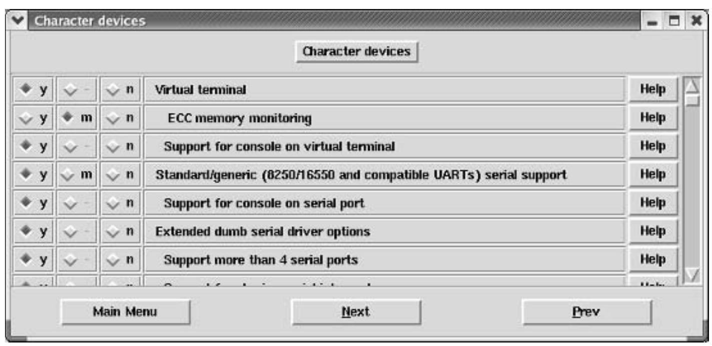


FIGURE 12.32
Character Devices menu



Character devices also include some surprising kernel settings, such as tape drives, graphics cards, mice, and joysticks. Several submenus are included, as follows:

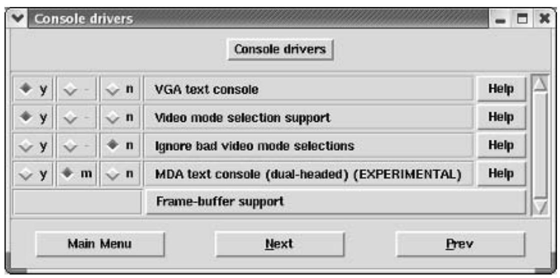
- I2C Support is a serial bus protocol required to support a wide variety of hardware, including Video For Linux kernel settings.
- Hardware Sensors Support includes a number of devices designed to monitor hardware; it's based on the work of the Linux System Hardware Monitoring project at `www2.lm-sensors.nu/~lm78`.
- The Mice submenu allows you to configure support for basic pointing devices such as a mouse or touchpad.

- Joysticks relate to devices associated with the game port on a PC.
- Watchdog Cards are common with embedded devices; they're designed to force reboots if there is no input for some specified period of time.
- Ftape relates to older tape drives connected to the 34-pin floppy disk controller. It includes drivers for several makes and models.
- The PCMCIA Character Devices submenu lets you emulate serial ports.

Console Drivers

Console drivers are straightforward: they allow for consoles, or text-mode terminals, in a graphical screen. The Console Drivers menu is shown in Figure 12.33.

Figure 12.33
Console Drivers menu

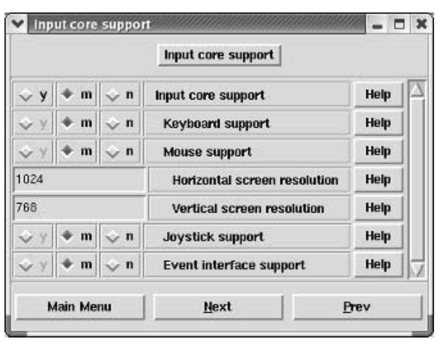


There is one submenu, Frame-Buffer Support. It allows applications to get to the graphical hardware through a buffer. It's experimental for Intel-based systems and generally is not required.

Input Core Support

Input core support is required for Human Interface Device (HID) interaction with Linux. An HID is a physical interface that sends signals to your computer, including keyboards, mice, and joysticks. The Input Core Support menu is shown in Figure 12.34.

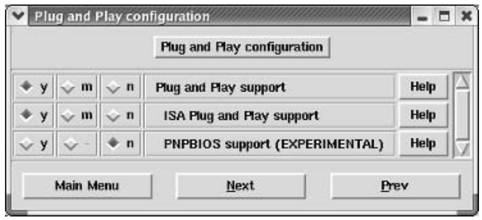
Figure 12.34
Input Core Support menu



Plug-and-Play Configuration

Linux plug-and-play support in the kernel is straightforward. As shown in Figure 12.35, you can activate basic plug-and-play support, as well as the special commands required for ISA plug-and-play devices.

Figure 12.35
Plug And Play Configuration menu



I2O Devices

I2O is the acronym for the Intelligent Input/Output architecture, which allows drivers to be split into modules for the hardware and operating system. I2O is commonly used with embedded devices; most users won't use or need to enable I2O devices in the Linux kernel. The I2O Device Support menu is shown in Figure 12.36.

Figure 12.36
I2O Device Support menu

Sound

Linux supports an impressive array of sound cards. While Linux does not support every sound card, you may be able to make some sound cards work by configuring an appropriate alternative, such as a SoundBlaster card. If you don't see a driver for your sound card in this menu, shown in Figure 12.37, check the documentation or consult the manufacturer of your sound card for advice.

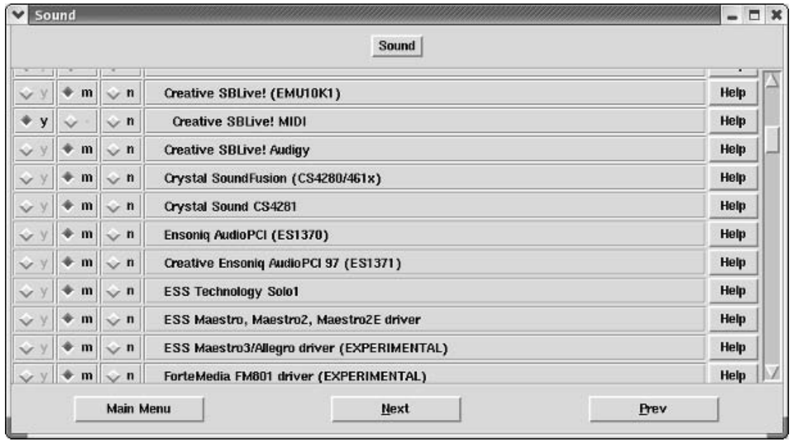


FIGURE 12.37
Sound menu

MULTIMEDIA

Closely related to sound is multimedia. The Multimedia Devices menu may not be quite what you'd expect. It includes a submenu for Video For Linux, which requires I2C serial support in the Character Devices menu. It also includes a submenu for Radio Adapters, which includes a list of regular radios that you can install on your computer. The Multimedia Devices menu is shown in Figure 12.38.

FIGURE 12.38
Multimedia Devices menu

Other Software Support

The remaining kernel menus are software menus that can't be classified into any of the other categories. They include basic interfaces for encryption, filesystems, load profiling, kernel debugging, and libraries.

CRYPTOGRAPHY

As of this writing, you can't configure these options through the `xconfig` menu; the options are blank for the source code associated with kernel 2.4.21-9.EL. However, you can configure it using the `make menuconfig` menus described earlier. Navigate to the Cryptographic Options submenu shown in Figure 12.39. This allows you to use several forms of strong encryption on Linux.

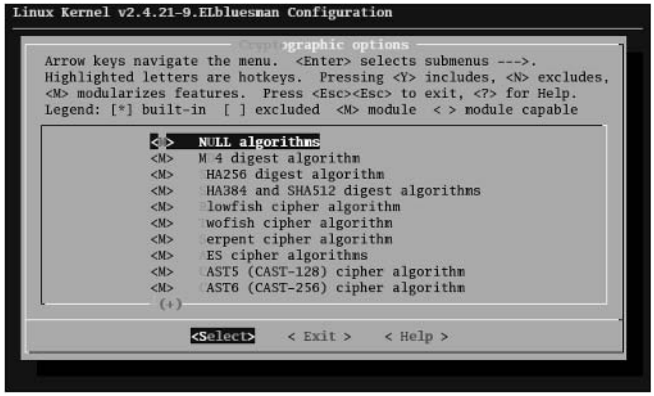


FIGURE 12.39
Cryptographic Options menu

FILESYSTEMS

The Linux kernel File Systems menu allows you to configure the types of formats Linux can read, as well as quotas on each partition. Linux supports a number of filesystem formats, including many you're familiar with from Chapter 7. The File Systems menu is shown in Figure 12.40.

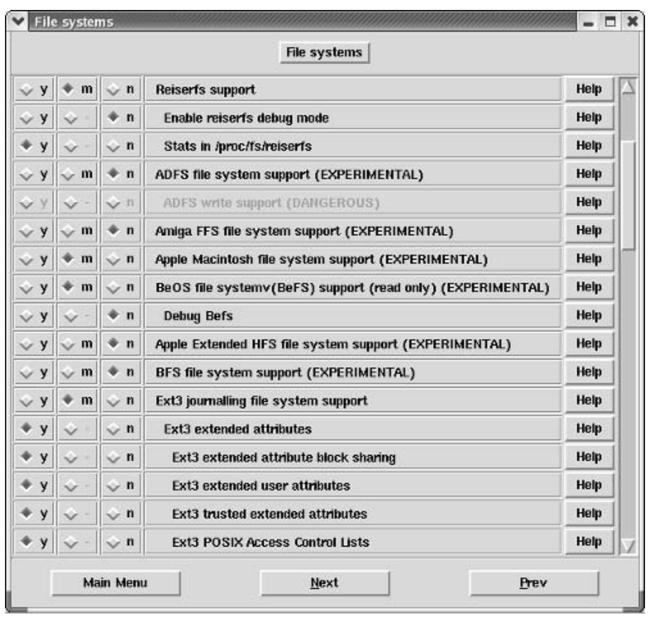


FIGURE 12.40
File Systems menu

Be careful. Linux support for several filesystems is experimental. This includes the module that lets you write a file to a Microsoft NTFS style filesystem, which is labeled as "DANGEROUS."

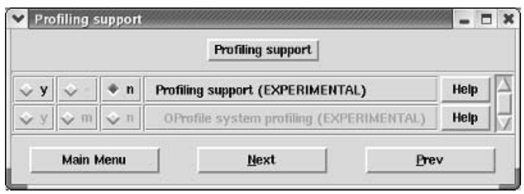
NOTE *The terms* file systems *and* filesystems *are used interchangeably.*

WARNING *I would not activate the NTFS Write Support module. The associated help file suggests you "...back up your NTFS volume first, since it will probably get damaged." However, you can get support based on the work of the Linux NTFS project, at* `linux-ntfs.sourceforge.net`.

Profiling

The latest Linux kernels are incorporating support for profiling the performance of your system. It's based on the OProfile system described at `http://oprofile.sourceforge.net/about.php3`; it is currently "Alpha-level" experimental software. However, it can be useful for tracking system performance. This menu is not active if you've deactivated the development drivers setting in the Code Maturity Level Options menu. The Profiling Support menu is shown in Figure 12.41.

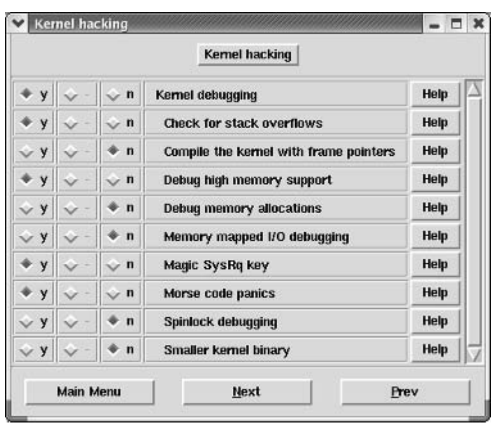
FIGURE 12.41
Profiling Support menu



Kernel Hacking

The Kernel Hacking menu, shown in Figure 12.42, supports drivers that can help you debug driver or other kernel problems. This menu is generally used by developers.

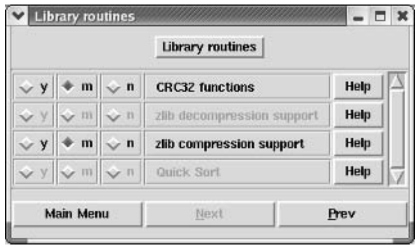
FIGURE 12.42
Kernel Hacking menu



Library Routines

The Library Routines menu shown in Figure 12.43 includes CRC32 checksum support, `zlib` compression and decompression support for data streams, and the quick sort data algorithms.

Figure 12.43 Library Routines menu



Remember, once you've completed your changes to the kernel configuration, you'll still need to satisfy dependencies and more, as described earlier in this chapter. This is also summarized at the end of the chapter; the next thing you should do is step 6, which is to run the `make dep` command.

Updating the Bootloader

If you've run the `make install` command, Linux should have updated your bootloader, as well as the files in your `/boot` directory, automatically. You can inspect the results for yourself; you should see files with similar names in your `/boot` directory, using the `EXTRAVERSION` variable as defined in your `/usr/src/linux-2.4/Makefile`.

While GRUB is the default bootloader for Red Hat Enterprise Linux, LILO is still in fairly common use. When you recompile a kernel, you should set up your bootloader to boot from either kernel, as though they were two distinct operating systems. You can check the result in your bootloader.

Inspecting GRUB

Assuming GRUB is your bootloader, open `/etc/grub.conf` in the text editor of your choice. If Red Hat Enterprise Linux is the only operating system on your computer, the key commands are as follows:

```
default=0
title Red Hat Enterprise Linux (2.4.21-4.EL)
   root (hd0,0)
   kernel /vmlinuz-2.4.21-4.EL ro root=LABEL=/
   initrd /initrd-2.4.21-4.EL.img
```

Now take the kernel you just recompiled. The main compressed kernel file is `vmlinuz-2.4.21-4.ELbluesman`; the corresponding Initial RAM file is `initrd-2.4.21-4.ELbluesman.img`. Since you've installed these files in the same `/boot` directory, none of the other parameters will change. The previously described `make install` command should have already added a stanza with the newly compiled kernel:

```
default=1
title Red Hat Enterprise Linux (2.4.21-4.ELbluesman)
```

```
        root (hd0,0)
        kernel /vmlinuz-2.4.21-4.ELbluesman ro root=LABEL=/
        initrd /initrd-2.4.21-4.ELbluesman.img
    title Red Hat Enterprise Linux (2.4.21-4.EL)
        root (hd0,0)
        kernel /vmlinuz-2.4.21-4.EL ro root=LABEL=/
        initrd /initrd-2.4.21-4.EL.img
```

Remember, nothing more is required. When you reboot your computer, you'll see both titles in the GRUB menu, as shown in Figure 12.44. Since `default=1`, the old kernel in the first stanza is still the default. We described a similar version of `grub.conf` in Figure 12.2. For a detailed analysis of GRUB, see Chapter 11.

FIGURE 12.44
Revised GRUB

Inspecting LILO

If you use LILO as your bootloader, open `/etc/lilo.conf` in the text editor of your choice. If Red Hat Enterprise Linux is the only operating system on your computer, the key commands are as follows:

```
default=2.4.21-4.EL
image=/boot/vmlinuz-2.4.21-4.EL
     label=2.4.21-4.EL
     initrd=/boot/initrd-2.4.21-4.EL.img
     read-only
     append="root=LABEL=/"
```

Now take the kernel you just recompiled. The main compressed kernel file is `vmlinuz-2.4.21-4.ELbluesman`; the corresponding Initial RAM file is `initrd-2.4.21-4.ELbluesman.img`. Since you've installed these files in the same `/boot` directory, none of the other parameters will change. The

previously described `make install` command should have already added a stanza with the newly compiled kernel:

```
default=2.4.21-4.EL
image=/boot/vmlinuz-2.4.21-4.ELbluesman
     label=2.4.21-4.ELblue
     initrd=/boot/initrd-2.4.21-4bluesman.img
     read-only
     append="root=LABEL=/"
image=/boot/vmlinuz-2.4.21-4.EL
     label=2.4.21-4.EL
     initrd=/boot/initrd-2.4.21-4.EL.img
     read-only
     append="root=LABEL=/"
```

You may note that Linux has abbreviated the version number in the label for the kernel we customized, version `2.4.21-4.ELbluesman`. As you can see in Figure 12.45, the label is reflected in the revised version of the LILO bootloader.

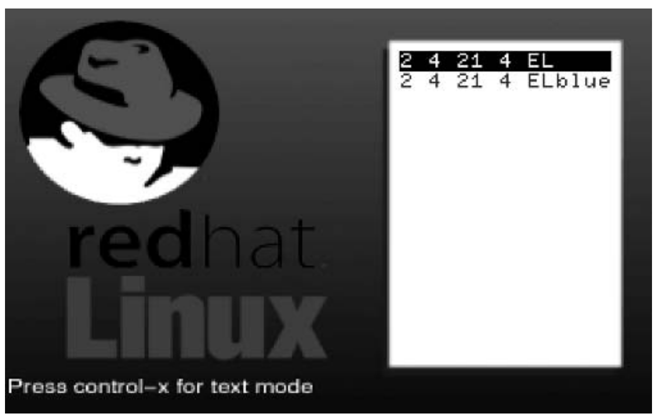


FIGURE 12.45
Revised LILO

Save your changes. With LILO, you need to run the `lilo` command to write the changes to the Master Boot Record of your hard disk (You may need to install the `lilo` RPM). Since the `default` setting is `2.4.21-4.EL`, LILO will still automatically boot your old kernel unless you specifically select the new one in the LILO boot menu.

If you're experimenting with LILO and GRUB, make sure you know the hard drive device with your `/boot` directory. For example, if it's `/dev/hda`, you can reinstall GRUB with the `grub-install /dev/hda` command.

Summary

The idea of upgrading and recompiling the Linux kernel strikes fear into many. While the steps are labor-intensive, there is nothing difficult about this process.

The easiest way to upgrade a kernel is to install a newer Red Hat Enterprise Linux kernel RPM package. When installed and not upgraded, a new kernel automatically upgrades the bootloader as well. Alternatively, if the upgrade is small, you can download and install a patch.

If you want to change the configuration of a kernel, the process is long. This is a summary of the basic steps:

1. Download the source code for the new kernel, preferably the Red Hat Enterprise `kernel-source` RPMs.
2. Install the RPMs associated with kernel tools such as `menuconfig` or `xconfig`. It may be faster to install the Development Tools package group using the GUI Package Management tool.
3. Navigate to the directory with your kernel source code. Select a value for `EXTRAVERSION` in the `Makefile`. Back up any current hidden `.config` file. Clean the current source code with the `make mrproper` command.
4. Use a baseline configuration; some are available in `/boot`, others in the `configs` subdirectory. Alternatively, you could use the local `.config` file or create one with the `make oldconfig` command.
5. Open a kernel configuration editor using `make menuconfig` or `make xconfig`. Make your changes, and save.
6. Set up the dependencies with the `make dep` command.
7. Run the `make clean` command to prepare the revised files to build your new customized kernel.
8. Create a compressed kernel image with the `make bzImage` command. Note the directory with the image.
9. Organize your kernel modules with `make modules` and `make modules_install`.
10. Finish the process with the `make install` command. This creates an Initial RAM disk for your new kernel and copies the needed files to the `/boot` directory. It also updates the default bootloader with the name and location of these files.

In the next chapter, we'll pick up with other administrative functions. Job managers such as `cron` and `at` allow administrators to run programs on an automated basis. Other key administrative skills include log file analysis and service management.

Chapter 13

The Administrative Nitty-Gritty

Administering computers can be a complicated job. Even in small organizations, there are users and groups to configure, backups to create, databases to maintain, and similar chores. Many administrative jobs are time-consuming exercises. If you run them during the day, they can overwhelm a system that's already trying to keep up with your users.

You could change your hours and run these jobs at night. But what if you're responsible for several facilities? Even Linux administrators deserve a personal life.

To support these tasks, Linux includes the `at` and `cron` daemons, which help you automate the tasks you need to run, any time, on any schedule. While `at` is a onetime management tool, `cron` allows you to set up jobs to run on regular schedules. We'll show you the advantages of downloading `anacron` to make sure your scheduled jobs run on a regular basis.

If you don't have an immediate solution, the first place to start troubleshooting is with the log files. Linux logs—most of which are located in the `/var/log` directory—are a rich source of information on the activity of your system. Different log files can help you monitor security, login activity, daemon status, and more.

As a Linux administrator, you should be familiar with a number of basic commands. The `ps`, `top`, and `kill` commands help you manage processes. You can check current logins with `who`. The `nice` and `renice` commands help you prioritize what's running. The `nohup` command can also help you run commands even after logging out of your account.

There are a couple of related configuration tools for tuning the kernel and for automating time synchronization on your computer. This chapter covers the following topics:

- Using the `cron` daemon
- Using the `at` daemon
- Service management tools
- Troubleshooting with logs
- Process management
- Using related configuration tools

Using the *cron* Daemon

If you were a computer operating system and did not need sleep, you could back up users' files at night. You could also rotate logs and delete temporary files while others sleep.

The `cron` daemon (also known by its command script, `crond`) performs these tasks on an automated basis. When Linux starts, it runs `crond` as a background process. Every minute, it checks the appropriate configuration files to see if something needs to be run.

There are two groups of `cron` configuration files. One group is governed by a global configuration file, `/etc/crontab`. Another is based on those created by individual users with the `crontab` command.

Formatting *cron*

To understand how `cron` works, it's best to start with the basic `cron` configuration file, `/etc/crontab`. This file specifies several environment variables, including SHELL, PATH, and HOME. The following is a line-by-line analysis of this file:

```
SHELL=/bin/bash
```

The commands in this file are based on the bash shell.

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

When the commands in this file are located in the noted directories, the full directory path is not required. The PATH in `/etc/crontab` also determines the order in which the directories are searched. For example, if the `flight` command exists in both the `/sbin` and `/usr/bin` directories, `cron` runs the `/sbin/flight` command.

```
MAILTO=root
```

Every time `crond` actually does something, notification is mailed to the root user.

```
HOME=/
```

The home directory associated with this `/etc/crontab` configuration file is the root (`/`) directory.

```
# run-parts
```

While this is a comment, the `run-parts` command is included in the following four lines. It runs every script file in the specified directory. This allows you to organize the scripts you need to run on a periodic basis.

```
01 * * * * root run-parts /etc/cron.hourly
```

The following command runs every script in the `/etc/cron.hourly` directory at one minute past every hour, every day:

```
02 4 * * * root run-parts /etc/cron.daily
```

The following command runs every script in the `/etc/cron.daily` directory at 4:02 A.M. every day:

```
22 4 * * 0 root run-parts /etc/cron.weekly
```

The following command runs every script in the `/etc/cron.weekly` directory at 4:22 A.M. every Sunday.

```
42 4 1 * * root run-parts /etc/cron.monthly
```

The following command runs every script in the `/etc/cron.monthly` directory at 4:42 A.M. on the first day of every month

The numbers and asterisks in the commands may seem cryptic. Let's take a closer look.

The Syntax of *cron*

To use `cron` effectively, you need to understand the time and date fields on the left side of each command in a `cron` file. Table 13.1 shows the five fields, from left to right.

TABLE 13.1: *CRON* FIELDS

FIELD	ALLOWABLE RANGE
Minute	0–59
Hour	0–23, where 0 is midnight and 20 is 8 P.M.
Day	1–31
Month	1–12
Day of week	0–7, where 0 and 7 both represent Sunday

An asterisk in any field is a wildcard. For example, if the first field contains an asterisk, that particular job runs every allowable minute.

If you want to specify a range such as every hour between 8:00 A.M. and 4:00 P.M., set the second field to `8-16`. Alternatively, you can run a job every other day by setting the third field to `*/2`. As you can see, once you know each of the five fields (minute, hour, day, month, day of week), there's nothing cryptic about any of the `cron` command fields.

Standard *cron* Jobs

When you install Red Hat Enterprise Linux, the standard configuration includes a set of `cron` jobs. This configuration allows you to organize `cron` jobs on an hourly, daily, monthly, and weekly basis. Each of these categories includes its own directory: `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, and `/etc/cron.monthly`.

The following are several standard `cron` jobs that run on a daily basis:

logrotate Rotates logs periodically. For example, Red Hat Enterprise Linux rotates five weeks of logs, and the `/var/log/messages` entries from the previous week are kept in the `/var/log/messages.1` file.

slocate.cron Refreshes the database associated with the `locate` command. By default, the database updates exclude directories that are networked from other computers, as well as several temporary directories.

tmpwatch Deletes files in the `/tmp` and `/var/tmp` directories. By default, files in these directories are deleted if they haven't been accessed in 240 and 720 hours, respectively.

User *cron* Jobs

Linux users may want to schedule their own `cron` jobs. For example, someone may want to manage a database in the middle of the night. As long as that user is not on the `/etc/cron.deny` list (described later in this chapter), that user can start his or her own `cron` file by using the `crontab -e` command.

NOTE *While* `crontab` *uses the* `vi` *editor by default, you can set it to use another editor. For example, if you want to use* `emacs` *to edit your* `cron` *file, run the* `export EDITOR=emacs` *command.*

For example, assume you've configured a script named `goodback` to back up all the files in your home directory. You want to run `goodback` every Sunday morning at 1:36 A.M. Assume your username is ez, and your script is in your a default home directory (`/home/ez`). Log in as ez, and then run `crontab -e`. Assuming you're using the default `vi` editor, type **i** to enter insert mode, and type the following line:

```
36 1 * * 0     /home/ez/goodback
```

Once you've saved the file, you can check the contents with the `crontab -l` command. All user `cron` files are stored in the `/var/spool/cron` directory and are accessible by default to the owner and the root user.

NOTE *If you're creating a* `cron` *file, you should also assign the SHELL, PATH, and HOME variables. It's also a good idea to set the MAILTO variable, as it can notify you whenever* `cron` *actually runs one of your jobs. For guidance, see the earlier section, "Formatting* `cron`*," which detailed the default* `/etc/crontab` *file.*

SCRIPT MANAGEMENT

When you run a `cron` job, you're running a script. This is an executable file with commands that you could otherwise run at the command-line interface. You can also put any command you use frequently into a file by using a text editor. Save the file, and then use the `chmod +x` *`script1`* command to make it executable. Assuming, for example, the file is in the `/path/to` directory, you run it at any time by typing the `/path/to/`*`script1`* command. If you have several commands you normally run at the same time, you could expand that one-line file to include several commands. This is a great timesaver.

Saving your scripts to a directory in your PATH is even more efficient. For example, say your username is tb. Run the `echo $PATH` command. You should see the `/home/tb/bin` directory in your PATH. If you save scripts such as *`script1`* to `/home/tb/bin`, all you'd need to do to run that script is run the *`script1`* command.

Just remember, if you're going to run *`script1`* as a `cron` job, you need to add the appropriate directory to the PATH, as described earlier.

cron Security

By default, `cron` tools are available to all users. You can limit access to `cron` by creating `/etc/cron.allow` and/or `/etc/cron.deny` files. The following are three possible scenarios for these files:

- Neither of these files exists, which means every user is allowed access to `cron`.
- Users listed in `/etc/cron.allow` are the only ones allowed access to `cron` tools. If you also have an `/etc/cron.deny` file, it is ignored.
- Users listed in `/etc/cron.deny` are not allowed to use `cron` tools. This assumes `/etc/cron.allow` does not exist.

Adding *anacron*

Red Hat Enterprise Linux 3 is an excellent operating system, reliable and scalable. However, when Red Hat decided to scale down the number of RPM packages from about 1,500 in Red Hat Linux 9 to 1,100 in Red Hat Enterprise Linux 3, it left out a few packages I consider excellent for server applications. One of these packages is `anacron`.

If your server is always powered on, Red Hat is correct—you don't need `anacron`. This software is designed to run `cron` jobs that didn't get a chance to run when your servers were powered down.

If you power down your servers on a regular schedule, you may not need `anacron`; all you need to do is adjust the start times described earlier in the `/etc/crontab` configuration file.

But in many situations you may be running Red Hat Enterprise Linux on a server that's powered down at irregular intervals. You may be running it on a workstation or even (like me) on a laptop computer. You may be in an environment where the power supply is less than reliable. In these cases, `anacron` is for you.

NOTE *Apparently, a number of Red Hat Enterprise Linux 3 users want* `anacron`*, too; see bug 103691 at* `bugzilla.redhat.com` *for more information.*

I've downloaded and run the Red Hat Linux 9 version of `anacron` on my Red Hat Enterprise Linux 3 computer. Remember, they share many identical packages. Red Hat Linux 9 software is available online from the Red Hat FTP site (`ftp.redhat.com`) or one of the mirrors listed at `www.redhat.com/download/mirror.html`. Once downloaded, you can install it with the `rpm` command; for example, if you've downloaded it to `/tmp`, you can do so with the following command:

```
# rpm -Uvh /tmp/anacron-*
```

The `anacron` package includes the `/etc/anacrontab` file, which is similar to the previously described `/etc/crontab` file. Essentially, when your system is booted, it checks the daily, weekly, and monthly `cron` jobs. For example, the following line from the Red Hat Linux 9–based `/etc/anacrontab` file:

```
7    70     cron.weekly     run-parts /etc/cron.weekly
```

checks the jobs in the `/etc/cron.weekly` directory. If they have not been run in 7 days, it runs those jobs 70 minutes after Linux starts the `anacron` daemon.

Using the *at* Daemon

One of the drawbacks of a `cron` job is that it is scheduled to be run on a regular basis. Sometimes, you just want to run a specific task once and then forget it. That's where the `at` daemon comes in.

It's easy to set up an `at` job. You can specify the time when you want to run the program, or you can use the associated `batch` command to start the job when your computer is relatively free.

This daemon works more like the print process; jobs are spooled in the `/var/spool/at` directory and executed at the desired time.

Setting Up an *at* Job

The `at` daemon works almost as if it were a separate shell. When you run the `at` *`time`* command, it sends you to a command prompt where you can enter the commands and programs of your choice. The `at now +` *`time`* command works as well; the job runs after the specified time period has passed.

For example, assume you're working on a large database and want to process the data when nobody else is using the system, say at 2:05 A.M. You've set up the `/home/mj/airplane` script to manage your database, and you plan to process the results in the `/home/mj/air-safe` file. The normal way to do this is with the following commands:

```
# at 2:05 tomorrow
at> /home/mj/airplane > /home/mj/air-safe
at> Ctrl-D
```

You have a number of different ways to set up the time in the `at +` *`time`* command, as shown in Table 13.2.

TABLE 13.2: *AT* DAEMON TIME PARAMETERS

PERIOD	EXAMPLE	COMMENT
Minute	`at now + 5 minutes`	The jobs will start in five minutes.
Hour	`at now + 1 hour`	The jobs will start in one hour.
Days	`at now + 3 days`	The jobs will start in three days.
Weeks	`at now + 2 weeks`	The jobs will start in two weeks.
Fixed	`at midnight`	The jobs will start at midnight.
Fixed	`at 10:30pm`	The jobs will start at 10:30 P.M.
Fixed	`at 1:00 10/12/04`	The jobs will start on October 12, 2004, at 1 A.M.

Job Queue

Once you've entered a job, you can make sure it's in the queue by using the `atq` command. As you can see, the output gives you the job number, the responsible user, and the time when the job is to be executed. The letter before the username (`a` or `b`) indicates whether it's an `at` or a `batch` job.

```
# atq
8    2003-03-08 02:05 a mj
```

It's easy to remove a job. Just use the `atrm` *`jobnumber`* command. For example, the following command deletes job 8 from the queue:

```
# atrm 8
```

Batch Jobs

The `batch` command is a specialized version of `at` that runs `at` jobs. By default, jobs created with this command run only when the demand on your CPU is less than 80 percent of its capacity.

The `batch` command is equivalent to the `at -q b` command.

Security

Similar to the `cron` daemon, `at` uses the `/etc/at.allow` and `/etc/at.deny` files to regulate access to this system. By default, Red Hat Enterprise Linux installs a blank `/etc/at.deny` file. This allows all users access to the `at` system.

As long as the `/etc/at.allow` file does not exist, only the users listed in `/etc/at.deny` are denied use to `at`. If you add users to `/etc/at.allow`, only those users are allowed to use the `at` command. In this case, the `/etc/at.deny` file is ignored.

Service Management Tools

One key skill for Linux system administrators is service management. You can start and stop current services with the scripts in the `/etc/rc.d/init.d` directory. In addition, you can ensure that the services of your choice are active only at specific runlevels.

/etc/rc.d/init.d Scripts

The services you install in Red Hat Enterprise Linux have their own scripts in the `/etc/rc.d/init.d` directory. It's likely that you have a substantial number of scripts on your system; Figure 13.1 shows a sample from my desktop computer.

Take a look at some of these scripts. Open them in a text editor. Near the end of each script, you should see a series of commands similar to Figure 13.2. This particular script, `smb`, manages Samba. Some of the commands in different scripts do vary.

FIGURE 13.1
Service scripts in */etc/rc.d/init.d*

```
[root@Enterprise3 root]# \ls /etc/rc.d/init.d/
aep1000       dhcrelay    kdcrotate      network     rwhod         tux
anacron       firstboot   keytable       nfs         saslauthd     vncserver
apmd          functions   killall        nfslock     sendmail      vsftpd
arptables_jf  gpm         kudzu          nscd        single        winbind
atd           halt        lisa           ntpd        smartd        xfs
autofs        hpoj        mdmonitor      pcmcia      smb           xinetd
bcm5820       httpd       mdmpd          portmap     snmpd         ypbind
crond         innd        microcode_ctl  postfix     snmptrapd     yppasswdd
cups          ip6tables   named          psacct      spamassassin  ypserv
dc_client     iptables    netdump        random      squid         ypxfrd
dc_server     irda        netdump-server rawdevices  sshd          yum
dhcpd         irqbalance  netfs          rhnsd       syslog
[root@Enterprise3 root]#
```

FIGURE 13.2
The innards of a service script

```
# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/dhcpd ]; then
            stop
            start
            RETVAL=$?
        fi
        ;;
  configtest)
        dhcpd -t
        RETVAL=$?
        ;;
  status)
        status dhcpd
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|condrestart|configtest|status}"
        exit 1
esac
```

As you can see Figure 13.2, you can run several actions for that particular service, as shown in Table 13.3. The table simply reflects the actions shown in Figure 13.2; the actions associated with a different script will vary.

TABLE 13.3: SERVICE SCRIPT ACTIONS

ACTION	DESCRIPTION
`start`	Starts the service; equivalent to the `service` *`script`* `start` command.
`stop`	Starts the service; equivalent to the `service` *`script`* `stop` command.
`restart`	Shuts down the service, then starts it again; equivalent to the `service` *`script`* `restart` command.
`reload`	Makes the service reread any applicable configuration files without restarting; equivalent to the `service` *`script`* `reload` command (some scripts require a restart to reread configuration files).
`condrestart`	If the service is "locked," this switch shuts down the service and then starts it again. Equivalent to the `service` *`script`* `condrestart` command.
`configtest`	Tests the configuration file, usually for syntax.
`status`	Provides the current status of the service; equivalent to the `service` *`script`* `status` command.

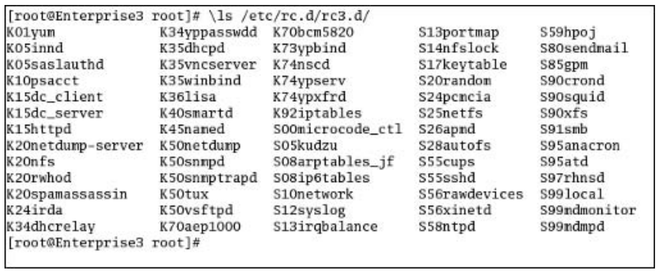
For example, if you wanted to restart Samba, you could run one of the following two commands as the root user:

```
# /etc/rc.d/init.d/smb restart
# service smb restart
```

Activation at Different Runlevels

You can make a service start and stop at different runlevels. For example, take a look at Figure 13.3.

FIGURE 13.3
Services at runlevel 3



```
[root@Enterprise3 root]# \ls /etc/rc.d/rc3.d/
K01yum             K34yppasswdd  K70bcm5820        S13portmap      S59hpoj
K05innd            K35dhcpd      K73ypbind         S14nfslock      S80sendmail
K05saslauthd       K35vncserver  K74nscd           S17keytable     S85gpm
K10psacct          K35winbind    K74ypserv         S20random       S90crond
K15dc_client       K36lisa       K74ypxfrd         S24pcmcia       S90squid
K15dc_server       K40smartd     K92iptables       S25netfs        S90xfs
K15httpd           K45named      S00microcode_ctl  S26apmd         S91smb
K20netdump-server  K50netdump    S05kudzu          S28autofs       S95anacron
K20nfs             K50snmpd      S08arptables_jf   S55cups         S95atd
K20rwhod           K50snmptrapd  S08ip6tables      S55sshd         S97rhnsd
K20spamassassin    K50tux        S10network        S56rawdevices   S99local
K24irda            K50vsftpd     S12syslog         S56xinetd       S99mdmonitor
K34dhcrelay        K70aep1000    S13irqbalance     S58ntpd         S99mdmpd
[root@Enterprise3 root]#
```

You can see from the `/etc/rc.d/rc3.d` directory that Apache is killed when Linux starts runlevel 3 (`K15httpd`). If you want to start that service at runlevel 3, you need to change it into a start script. The standard method is the `chkconfig` command. To list the current runlevels associated with Apache (`httpd`), run the following command:

```
# chkconfig --list httpd
httpd   0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

The output shows that Apache isn't started at any runlevel. To make sure it starts the next time you boot Linux, you need to activate it at the desired runlevels. For example, the following command starts Apache at runlevels 3 and 5, in standard multiuser and GUI modes:

```
# chkconfig --level 35 httpd on
```

You can confirm the effect by listing the files at runlevels 3 and 5; in this case, you'll see the `S85httpd` start script in each directory (`/etc/rc.d/rc3.d` and `/etc/rc.d/rc5.d`). Alternatively, just run the following command:

```
# chkconfig --list httpd
httpd   0:off  1:off  2:off  3:on  4:off  5:on  6:off
```

Service Configuration Tools

If you're not comfortable with the `chkconfig` command and need GUI tools, Linux offers two. Red Hat includes the Service Configuration tool, which you can start only in the GUI. There's also the `ntsysv` utility, which you can also start in the text console.

The Service Configuration tool is straightforward, though limited; it allows you to configure services only in runlevels 3, 4, and 5. (While runlevel 4 is not used by Red Hat, it is available for special configurations you may create.)

To start this tool, run the `redhat-config-services` command in a GUI, or click Main Menu ➢ System Settings ➢ Server Settings ➢ Services. This opens the basic menu shown in Figure 13.4.

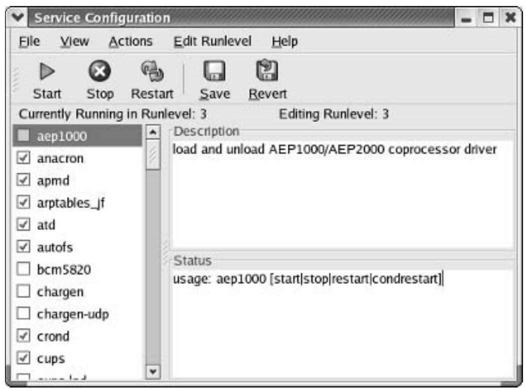


Figure 13.4
Service Configuration Tool

The service configuration window opens in the default runlevel as defined in `/etc/inittab`. Highlight the service of your choice, and you'll see a description of the associated daemon and its current status on the right side of the window.

In the Actions menu, you can start, stop, or restart a service; this corresponds to one of the following commands:

```
# service servicename stop
# service servicename start
# service servicename restart
```

In the Edit Runlevel menu, you can change the runlevel in work to 3, 4, or 5. Any changes you make are written to the `/etc/rc.d/rcn.d` directory, where *n* is the runlevel in question. Active daemons are associated with start scripts, and dormant daemons are associated with kill scripts. The next time you boot Linux, it reads the start and kill scripts in each of these directories.

If you're administering a server remotely, the GUI may not be available to you. You may not have installed a GUI. In either case, the `ntsysv` tool may be useful. For example, you can use `ntsysv --level runlevel` to view the services at different runlevels. For example, the `ntsysv --level 5` command could illustrate active services at runlevel 5, as shown in Figure 13.5. Brief descriptions are available for each service when you press F1.

FIGURE 13.5
Managing services with *ntsysv*

Troubleshooting with Logs

To paraphrase an old song associated with a restaurant in Berkeley, California, "You can have almost any log you want…in a Linux restaurant." The menu of available log files is impressive. You can configure logs by service or the severity of the problem.

You've already seen the workings of installation log files in Chapter 3. In the following sections, you'll review log files to see what happened with many Linux services.

Log files are governed by the syslog and kernel log daemons, `syslogd` and `klogd`, as configured in `/etc/syslog.conf`. Both daemons are active by default.

Log File Categories

You can use a system log file to diagnose a problem with installation, booting, specific services, and more. You can further divide logs into eight categories, listed here in descending order of importance:

- `emerg` (emergency)
- `alert`
- `crit` (critical)
- `err` (error)
- `warning`
- `notice`
- `info`
- `debug`

Log files are organized as described in the Linux `/etc/syslog.conf` configuration file. Take a look at the default Red Hat Enterprise Linux version of this file in Figure 13.6.

FIGURE 13.6
/etc/syslog.conf

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none      /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log

#
# INN
#
news.=crit                                        /var/log/news/news.crit
news.=err                                         /var/log/news/news.err
news.notice                                       /var/log/news/news.notice
~
```

As you can see, most logs are located in the `/var/log` directory. If you activate kernel messages, they are normally sent to the console (your screen). Some daemons, such as Internet Network News (`innd`), include additional specifications in this file.

Logs are maintained through a standard `cron` job, `logrotate`. As discussed earlier, it rotates log files on a weekly basis. Thus, the `/var/log/boot.log.1` file is from the previous week.

Take the first active line in this file, which specifies messages associated with several daemons. For example, the first statement, `*.info`, sends all messages of `info` level and higher (`notice`, `warning`, `err`, `crit`, `alert`, and `emerg`) to the appropriate log file.

System Logs

Now let's look at some of the system logs in the `/var/log` directory. The `dmesg` file consists of basic boot messages associated with starting Linux. The `message` file includes additional process messages after Linux boots on your computer. The `boot.log` file lists messages related to starting and stopping daemons. And `wtmp` helps you monitor logons.

GETTING THE REST OF THE *DMESG*

In Chapter 11, you learned about how `/var/log/dmesg` helps you determine whether Linux detected your hardware. There's one other critical item at the end of this file: whether Linux has properly mounted your filesystems and swap space. If the mount was successful, you should see messages similar to the following:

```
EXT3 FS 2.4-0.9.19, 19 August 2002 on ide0(3,1), internal journal
```

```
EXT3-fs: mounted filesystem with ordered data mode
kjournald starting. Commit interval 5 seconds
```

This tells you that Linux has successfully mounted an ext3 filesystem with an internal journal on a partition. The kjournal daemon (`kjournald`) does the actual work of keeping the filesystem journal up-to-date. You'll see additional lines like this for each Linux partition.

Other */var/log/messages*

Other messages associated with hardware and services are documented in `/var/log/messages`. The following excerpt illustrates a couple of examples:

```
Mar 21 06:46:56 Enterprise3 kernel: pcnet32.c:v1.27a 10.02.2002
➥tsbogend@alpha.franken.de
Mar 21 06:46:56 Enterprise3 kernel: PCI: Found IRQ 10 for device 00:11.0
Mar 21 06:46:56 Enterprise3 kernel: pcnet32: PCnet/PCI II 79C970A at
➥0x10e0, 00 0c 29 1c bb 76 assigned IRQ 10.
Mar 21 06:46:56 Enterprise3 kernel: eth0: registered as PCnet/PCI II
➥79C970A
Mar 21 06:46:56 Enterprise3 kernel: pcnet32: 1 cards_found.
Mar 21 18:38:39 Enterprise3 sshd(pam_unix)[3412]: session opened for
➥user root by (uid=0)
Mar 21 18:38:46 Enterprise3 sshd(pam_unix)[3412]: session closed for
➥user root
```

Each line in this file includes some basic characteristics—such as the date, time, hostname, and service associated with each message. If available, the username and process identifier are also listed.

You can see two important developments in the code. First, Red Hat Enterprise Linux has detected a pcnet32 Network Card during the boot process. Next, someone has successfully accessed the `Enterprise3` computer through `sshd`, the Secure Shell daemon. As you'll see in Chapter 18, `sshd` is a critical tool that can help you administer a computer remotely.

But you may have a security breach. If the noted login is not authorized, a cracker may have broken into your system. See Chapter 17 for techniques you can use to secure your Linux system.

Note *In the Linux world,* hackers *are good people who just want to create better software.* Crackers*, on the other hand, are people who try to break into your system.*

Analyzing the */var/log/boot.log*

When services or daemons start and stop, they are listed in `/var/log/boot.log`. Take the example shown in Figure 13.7. The first line shown is, in fact, the last message of a shutdown on December 1. The second message is the first daemon started when you boot Linux.

Some services are associated with other parameters. For example, the `keytable` parameter shown in Figure 13.7 loads the `keymap` associated with your keyboard. Another example is where the `ntpd` service starts to synchronize your computer system clock with a central time server.

FIGURE 13.7
boot.log

```
Dec  1 22:38:50 Enterprise3 syslog: klogd shutdown succeeded
Dec  2 12:02:18 Enterprise3 syslog: syslogd startup succeeded
Dec  2 12:02:18 Enterprise3 syslog: klogd startup succeeded
Dec  2 12:02:19 Enterprise3 irqbalance: irqbalance startup succeeded
Dec  2 12:02:20 Enterprise3 portmap: portmap startup succeeded
Dec  2 12:02:22 Enterprise3 nfslock: rpc.statd startup succeeded
Dec  2 12:02:23 Enterprise3 keytable: Loading keymap:
Dec  2 12:02:23 Enterprise3 keytable:
Dec  2 12:02:23 Enterprise3 keytable:
Dec  2 12:02:23 Enterprise3 rc: Starting keytable:  succeeded
Dec  2 12:02:24 Enterprise3 random: Initializing random number generator:  succe
eded
Dec  2 12:02:24 Enterprise3 rc: Starting pcmcia:  succeeded
Dec  2 12:02:26 Enterprise3 netfs: Mounting other filesystems:  succeeded
Dec  2 12:02:27 Enterprise3 apmd: apmd startup succeeded
Dec  2 12:02:29 Enterprise3 autofs: automount startup succeeded
Dec  2 12:02:41 Enterprise3 cups: cupsd startup succeeded
Dec  2 12:02:42 Enterprise3 sshd:  succeeded
Dec  2 12:02:44 Enterprise3 xinetd: xinetd startup succeeded
Dec  2 12:02:53 Enterprise3 ntpd:  succeeded
Dec  2 12:02:54 Enterprise3 ntpd: ntpd startup succeeded
Dec  2 12:02:59 Enterprise3 nfs: Starting NFS services:  succeeded
Dec  2 12:03:00 Enterprise3 nfs: rpc.rquotad startup succeeded
                                                            165,1          1%
```

DETECTING REMOTE LOGINS

Login records are kept in a database file, `/var/log/wtmp`. You can use the `utmpdump` command to make this file readable. Take a look at Figure 13.8; this is part of the output when I ran `utmpdump /var/log/wtmp`. Note the login from IP address 128.99.1.64. As you can see, some user named michael has logged in from a remote terminal (`ttyS0`). If you don't know this user or IP address, you may have a security problem.

WARNING *You should know the network IP address range for your LAN. If your network does not include some of the addresses shown in Figure 13.8 and you don't have any remote users, be alert. Someone may have tried to break into your system. Chapter 17 includes techniques designed to block logins from suspicious networks.*

FIGURE 13.8
Checking login activity

```
[7] [03414] [ts/1] [root    ] [pts/1        ] [192.168.1.21         ] [192.168.1.2
1    ] [Sun Mar 21 18:38:39 2004 EST]
[8] [03412] [    ] [        ] [pts/1        ] [                     ] [0.0.0.0
    ] [Sun Mar 21 18:38:45 2004 EST]
[8] [00000] [/0  ] [        ] [pts/0        ] [                     ] [0.0.0.0
    ] [Sun Mar 21 18:56:31 2004 EST]
[7] [03475] [/0  ] [root    ] [pts/0        ] [:0.0                 ] [0.0.0.0
    ] [Sun Mar 21 18:56:40 2004 EST]
[7] [03475] [/1  ] [root    ] [pts/1        ] [:0.0                 ] [0.0.0.0
    ] [Sun Mar 21 19:22:05 2004 EST]
[7] [03475] [/2  ] [root    ] [pts/2        ] [:0.0                 ] [0.0.0.0
    ] [Sun Mar 21 19:31:23 2004 EST]
[8] [00000] [/2  ] [        ] [pts/2        ] [                     ] [0.0.0.0
    ] [Sun Mar 21 19:31:47 2004 EST]
[8] [02097] [2   ] [        ] [tty2         ] [2.4.21-4.EL          ] [0.0.0.0
    ] [Sun Mar 21 19:31:57 2004 EST]
[5] [03635] [2   ] [        ] [             ] [2.4.21-4.EL          ] [0.0.0.0
    ] [Sun Mar 21 19:31:57 2004 EST]
[6] [03635] [2   ] [LOGIN   ] [tty2         ] [                     ] [0.0.0.0
    ] [Sun Mar 21 19:31:57 2004 EST]
[7] [03635] [2   ] [michael ] [ttyS0        ] [                     ] [128.99.1.64
        ] [Sun Mar 21 19:32:03 2004 EST]

                                                            217,80         Bot
```

Daemon Logs

Most Linux daemons, such as `crond`, `httpd`, and `smbd`, are configured with log files in the `/var/log` directory. Each log file can provide clues as to the success or failure of any particular service. A clean example is shown in Figure 13.9, a view of the `/var/log/cron` file.

FIGURE 13.9
/var/log/cron

```
Mar 20 01:30:00 Enterprise3 CROND[4604]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 01:40:01 Enterprise3 CROND[4610]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 01:50:00 Enterprise3 CROND[4613]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 02:00:00 Enterprise3 CROND[4616]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 02:01:00 Enterprise3 CROND[4619]: (root) CMD (run-parts /etc/cron.hourly)

Mar 20 02:10:00 Enterprise3 CROND[4629]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 02:20:00 Enterprise3 CROND[4634]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 02:30:00 Enterprise3 CROND[4639]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 02:40:00 Enterprise3 CROND[4642]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 02:50:00 Enterprise3 CROND[4645]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 03:00:00 Enterprise3 CROND[4648]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 03:01:01 Enterprise3 CROND[4651]: (root) CMD (run-parts /etc/cron.hourly)

Mar 20 03:10:00 Enterprise3 CROND[4661]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 03:20:00 Enterprise3 CROND[4669]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 03:30:00 Enterprise3 CROND[4673]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 03:40:00 Enterprise3 CROND[4676]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 03:50:00 Enterprise3 CROND[4679]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 04:00:00 Enterprise3 CROND[4684]: (root) CMD (/usr/lib/sa/sa1 1 1)
Mar 20 04:01:00 Enterprise3 CROND[4687]: (root) CMD (run-parts /etc/cron.hourly)

Mar 20 04:02:00 Enterprise3 CROND[4697]: (root) CMD (run-parts /etc/cron.daily)
                                                            2100,1        Bot
```

Figure 13.9 illustrates the date and time `cron` jobs were executed. You may recognize these as the standard `cron` jobs listed earlier in this chapter, run at the standard times specified in `/etc/crontab`. If you see different times—say, for running the `/etc/cron.daily` scripts—you've probably installed the `anacron` service described earlier.

The following excerpt from `/var/log/httpd/access_log` tells you about one of the clients for your web server; for example, that particular client used the Lynx web browser from the U.S. eastern time zone during standard time (-0500).

```
127.0.0.1 - -[23/Mar/2004:14:05:26 -0500] "GET / HTTP/1.1" 200 8735
➥ "http://127.0.0.1/" "ELinks (0.4.2; Linux; 0x0)"
```

The following excerpt (from `/var/log/samba/smbmount.log`) shows a connection to a Microsoft Windows share through Samba:

```
[2004/03/23 13:53:32, 0] client/smbmount.c:send_fs_socket(405) mount.smbfs:
➥ entering daemon mode for service \\bluesman\downloads, pid 5711
```

As you add more daemons to your Red Hat Enterprise Linux system, more log files will appear in the `/var/log` directory. However, log files don't have to be stored in `/var/log`; it's determined by the configuration files associated with each daemon.

Other Logs

The `/var/log` directory contains a number of other log files. As you add more services, more log files will appear. Therefore, Table 13.4 is not a comprehensive list. We've also omitted log files that we covered earlier in this chapter.

TABLE 13.4: */VAR/LOG* LOG FILES

FILE	FUNCTION
`cups`	Directory with print log files.
`gdm`	Directory with GNOME start log files.
`kdm.log`	KDE start log file.
`ksyms`	Exported kernel symbols, such as drivers and modules.
`lastlog`	Specifies last login time and location, based on the `lastlog -u` *`username`* command.
`maillog`	Anything related to mail servers, such as startup, shutdown, aliases, and errors related to `sendmail`.
`news`	A directory of log files related to the InterNetNews (INN) server.
`redhat-config-network`	Includes changes created using this tool.
`rpmpkgs`	Currently installed RPMs.
`scrollkeeper.log`	For documents, especially in GUIs.
`secure`	Anything related to secure connections, including `ssh` and `xinetd`.
`squid`	Directory with the Squid web proxy server log files.
`up2date`	Log of actions using the Red Hat Update Agent.
`xdm`	Last login via the X Display Manager.
`xferlog`	Lists installations and upgrades.
`XFree86*`	Various X start log files.

Configuring Remote Logs

In the enterprise, you may have to administer a group of servers. You could log into each server remotely using a tool such as SSH (see Chapter 18). Alternatively, you can set up logs to write to one specific server.

Remote logging is disabled by default in Red Hat Enterprise Linux. This keeps a random user from filling your system with endless streams of information. However, if your servers are already behind a firewall, remote logging can make good sense.

It's easy to enable remote logging. When you start the `syslogd` daemon, just use the `-r` switch. The standard is to run `syslogd` without a time stamp (`-m 0`); you'd start it with the following command:

```
# syslogd -m 0 -r
```

But you don't want to start individual daemons by hand every time you start your Linux computer; all you need to do is set it up with the appropriate start script. As described earlier in this chapter, these scripts are located in the `/etc/rc.d/init.d` directory. Open the `syslog` script (yes, without the `d`) in the text editor of your choice; the switches are listed with the `SYSLOGD_OPTIONS` variable, which you can change to the following:

```
SYSLOGD_OPTIONS="-m 0 -r"
```

While this logging server may already be inside your protected network, you can choose to protect it further on this computer. If you also have a firewall on this computer, you'll need to allow UDP information through port 514. For example, if your LAN is on the private IP network 192.168.1.0, you'd use the following command:

```
# iptables -A RH-Firewall-1-INPUT -t filter -p udp --dport 514
➥!192.168.1.0/24 -j DROP
```

This command modifies your firewall; it drops any messages from outside your LAN that goes through the logging port (514). We explain `iptables`, port, and firewall concepts in Chapter 17.

Of course, you'll need to configure the remote servers from where you want the logging information. If your logging server has an IP address of 192.168.1.13, you can add the following command to the remote `/etc/syslog.conf` configuration files.

```
*.* @192.168.1.13
```

Naturally, you can substitute the hostname of your logging server for the IP address.

GUI Logs

Red Hat includes a graphical viewer for standard log files, the System Logs tool shown in Figure 13.10. You can start it in a GUI with the `redhat-logviewer` command or from a GNOME desktop by selecting Main Menu ➢ System Tools ➢ System Logs.

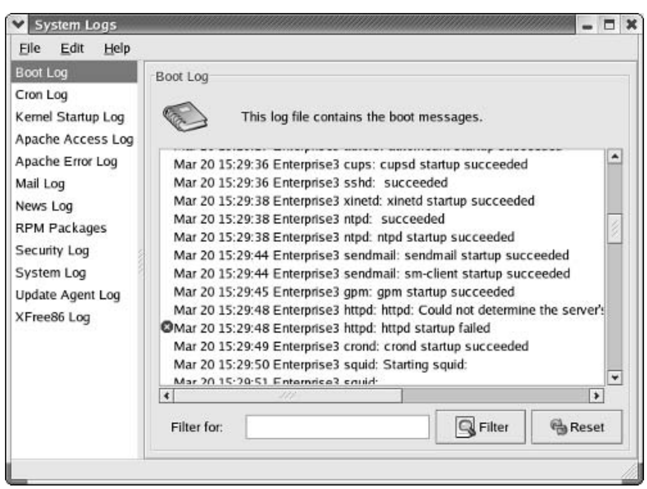


FIGURE 13.10 Reviewing system logs

Note the list of logs on the left and the view of the specific log file on the right. You can see right away, from the exclamation point (the alert icon) and "failed" messages, that there may be some problem with `httpd`, the Apache web server daemon.

You can use this tool to search for specific messages; enter the search term of your choice, and the System Logs tool isolates any messages with the search term. You may even realize that this search capability is a function of the `grep` command.

The `redhat-logviewer` is configured to review log files from standard locations. If you select Edit ➢ Preferences, that opens the Preferences dialog box, where you can change the file associated with a log and specify the messages that set off the alert icon.

Table 13.5 lists the standard locations for the `redhat-logviewer` log files.

TABLE 13.5: ***REDHAT-LOGVIEWER*** **STANDARD LOG FILE LOCATIONS**

Log Name	File Location
Boot	`/var/log/boot.log`
Cron	`/var/log/cron`
Kernel Startup	`/var/log/dmesg`
Apache Access	`/var/log/httpd/access_log`
Apache Error	`/var/log/httpd/error_log`
Mail	`/var/log/maillog`
News	`/var/log/spooler`
RPM Packages	`/var/log/rpmpkgs`
Security	`/var/log/secure`
System	`/var/log/messages`
Update Agent	`/var/log/up2date`
XFree86	`/var/log/XFree86.0.log`

If a log file is missing from the list, you may not have started the service previously. For example, if you don't see an Apache Access Log option in Figure 13.10, you probably haven't started or accessed the Apache web server on your computer.

Process Management

Anyone who manages a Linux computer needs to know how to manage processes.

Several key tools are available to help you manage Linux processes: `who`, `w`, and `ps`. These commands help you keep track of who is connected and what processes are being run, respectively. In addition, the `top` and `free` commands help you monitor the demands a service is placing on your computer. Finally, the `nohup` command can help you run another command and keep it going even after you log off your computer.

If any of your users are having a problem with any application, you can use the `kill` command to stop that application. If an important program or procedure is about to run, commands such as `nice` and `renice` can help you raise or lower the priority associated with the program of your choice.

Processes and *ps*

The `ps` command shows currently running processes or programs. When you type the `ps` command by itself, you see the processes associated with your setup. If you type the `ps aux` command, you can see everything running on your Linux system including daemons. Another useful variation is `ps l`, which returns a "long list" associated with each currently running process. Important categories from this command are shown in Table 13.6.

NOTE `ps` *is one of the few commands that does not require a dash in front of the associated switch.*

If you have a program that's out of control, you need the PID number to kill that problem program. Alternatively, if you need to run a program that's stuck waiting for CPU resources, you can use its PID to raise its priority.

TABLE 13.6: ***PS -AL*** **PROCESS CATEGORIES**

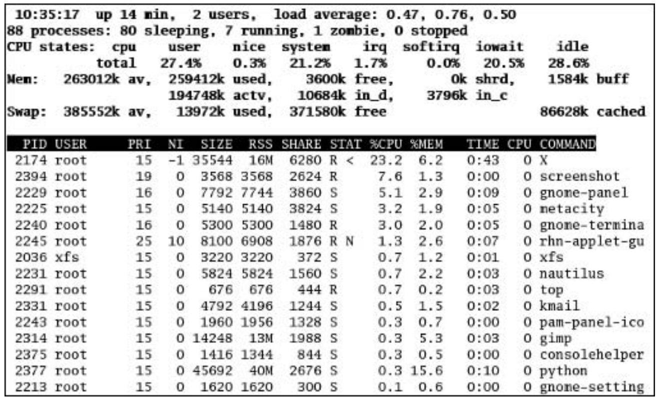
ITEM	EXPLANATION
PID	The process identifier. Every process is associated with a number known as a *process identifier*.
PPID	The parent process identifier. Every process has a parent except `init`. If you can't kill a process, you may be able to kill the parent process.
PRI	The priority value. Higher-priority programs get attention from your CPU more quickly. The highest-priority program has a PRI of –20. The lowest priority program has a PRI of 19.
S	The current status of the process. There are three options: Running (R), Sleeping (S), or Swapped (SW) to the swap partition.

Processes and memory with *top* and *free*

The `top` command helps you identify the programs that are "hogging" resources, specifically your CPU and RAM memory. For example, Figure 13.11 shows what the `top` command can see on a system with a less than ideal amount of RAM.

In this example, performance is slow, and you can hear the hard drive working constantly. The output shown in Figure 13.11 does not identify any specific application that has slowed the system. In the enterprise, with a substantial number of users, that is not a surprise. In any case, 256MB is a minimum, and is rather low for many server applications. It makes sense to add more RAM to this system.

However, it's a good idea to review this screen every now and then. If you're running a multiuser system, pay attention to users associated with troublesome processes. If their needs are legitimate, you have reason to add more RAM.



```
 10:35:17  up 14 min,  2 users,  load average: 0.47, 0.76, 0.50
88 processes: 80 sleeping, 7 running, 1 zombie, 0 stopped
CPU states:  cpu     user    nice  system    irq  softirq  iowait    idle
           total    27.4%    0.3%   21.2%   1.7%     0.0%   20.5%   28.6%
Mem:   263012k av,  259412k used,    3600k free,       0k shrd,    1584k buff
                    194748k actv,   10684k in_d,    3796k in_c
Swap:  385552k av,   13972k used,  371580k free                   86628k cached

  PID USER       PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME CPU COMMAND
 2174 root        15  -1 35544  16M  6280 R <  23.2  6.2   0:43   0 X
 2394 root        19   0  3568 3568  2624 R     7.6  1.3   0:00   0 screenshot
 2229 root        16   0  7792 7744  3860 S     5.1  2.9   0:09   0 gnome-panel
 2225 root        15   0  5140 5140  3824 S     3.2  1.9   0:05   0 metacity
 2240 root        16   0  5300 5300  1480 R     3.0  2.0   0:05   0 gnome-termina
 2245 root        25  10  8100 6908  1876 R N   1.3  2.6   0:07   0 rhn-applet-gu
 2036 xfs         15   0  3220 3220   372 S     0.7  1.2   0:01   0 xfs
 2231 root        15   0  5824 5824  1560 S     0.7  2.2   0:03   0 nautilus
 2291 root        15   0   676  676   444 R     0.7  0.2   0:03   0 top
 2331 root        15   0  4792 4196  1244 S     0.5  1.5   0:02   0 kmail
 2243 root        15   0  1960 1956  1328 S     0.3  0.7   0:00   0 pam-panel-ico
 2314 root        15   0 14248  13M  1988 S     0.3  5.3   0:03   0 gimp
 2375 root        15   0  1416 1344   844 S     0.3  0.5   0:00   0 consolehelper
 2377 root        15   0 45692  40M  2676 S     0.3 15.6   0:10   0 python
 2213 root        15   0  1620 1620   300 S     0.1  0.6   0:00   0 gnome-setting
```

FIGURE 13.11
top command output

Logins with *who* and *w*

As an administrator, you should check logons regularly; for example, the following output from `who` shows the same person logged on from two different locations:

```
mj      tty1    Mar 12 10:26
ywow    pts/1    Mar 12 10:27 (192.168.0.12)
mj      pts/0    Mar 12 10:41 (136.46.1.64)
```

Because user mj is logged on from the local computer and remotely from the computer at 136.46.1.64, you should be concerned that someone else is using mj's username and password to break into your system. You can actually get more information with the `w` command; in the same situation, you may see the following output:

```
 10:42:10  up 21 min, 3 users, load average: 0.26, 0.46, 0.48
USER    TTY   FROM           LOGIN@   IDLE   JCPU   PCPU   WHAT
mj      tty1  -             10:26am 15:21   5.01s  0.09s   /bin/sh
➥/usr/X11R6/bin/startx
ywow    pts/1 192.168.0.12  10:27am  0.00s  0.47s  0.06s   ls
mj      pts/0 136.46.1.64   10:41am  5.55   0.24s  0.24s   -bash
```

It's not much, but it shows that the user mj who is logged in from a remote system is not doing much at the moment; he's just in the bash shell. But if you see a program running from that remote system, pay attention.

Process *kill*

By reputation, Linux doesn't crash. There are reports of users and websites powered by Linux running without reboots for months at a time. One reason behind this is that system administrators can manage troublesome programs with the `kill` command.

For example, if a program such as Mozilla "locks up" on you while you're browsing the Internet, follow these steps to kill the program:

1. Open a command-line shell. If you can't open a command-line shell inside an X Window, start a new virtual console with the Ctrl+Alt+F*n* command, where *n* is a number between 1 and 6.
2. Run the `ps aux | grep mozilla` command. The number after your username is the PID of the process that is currently running Mozilla on your computer. Record that number. For purposes of this exercise, assume the number is 1789.
3. Run the `kill` *`PIDnumber`* command. Based on step 2, the actual command would be `kill 1789`. If the `kill` command doesn't work, run the `ps auxl | grep mozilla` command to find the PPID. You may need to kill those processes first.
4. As a last resort, use the `-9` switch, which kills the process even if it leaves other programs in your memory. In this case, you would use the `kill -9 1789` command.

nice and *renice*

The `nice` and `renice` commands let you run programs at different relative priorities. The priority of any program can range from –20 (highest) to 19 (lowest). The `nice` program starts another process with an adjusted priority. For example, you could set Mozilla to start after all others have finished by using the `nice -n 19 mozilla` command. If you have to focus Linux on one specific program, you need its PID. Once you find the program's PID (assume it's 1789 for this exercise), you can raise its priority with the `renice -10 1789` command.

NOTE *To understand priorities, keep in mind that* `nice` *and* `renice` *numbers seem reversed in Linux. If you want to make a program more important, use a negative number.*

Leaving a *nohup*

If you can't run a program with the priority you want, the `nohup` command can help. With `nohup`, you run a long command just before leaving your computer. For example, say you want to record an `.iso` file to a CD. You know that CDs take some time to record, but you need to pick up your child from school right now.

If your computer includes a CD recorder, the `nohup` command can help. If you want to take the `redhatcd1.iso` file and record it on a blank writeable CD, run the following command and log out of your user account, and the CD recording process will proceed automatically. Messages are written to the `nohup.out` file in the local directory.

```
# nohup cdrecord -v speed=4 dev=0,0,0 redhatcd1.iso
```

This assumes, of course, that you don't shut down Linux on your computer. More information on the `cdrecord` command is available in the next chapter.

Using Related Configuration Tools

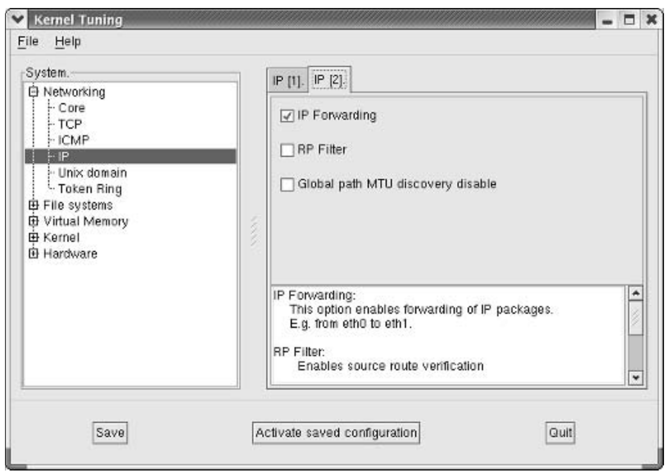
There are a couple of additional GUI tools of interest. While "real" Linux administrators would never resort to tools like these, administrators who are more familiar with Microsoft Windows can use these tools to help ease the transition. We cover the Kernel; and Date and Time configuration tools in the following sections, as they're administrative tools not covered in other parts of this book.

Tuning the Kernel

You can tune your kernel by adding commands to the `/etc/sysctl.conf` file. Alternatively, you can use the Red Hat Kernel Tuning tool by running `redhat-config-proc`. Both options allow you to modify settings in the `/proc` directory. Chapter 11 describes some of the files in this directory in greater detail. As of this writing, you can start this utility only from a GUI command-line interface; there's no entry in the GNOME Main Menu. Figure 13.12 displays the Kernel Tuning window.

WARNING *Be careful before you use* `redhat-config-proc`*. At the least, back up your current* `/etc/sysctl.conf` *file first. Any changes you make can change the functionality of your kernel, which could easily stop Linux from working.*

FIGURE 13.12
Kernel Tuning window



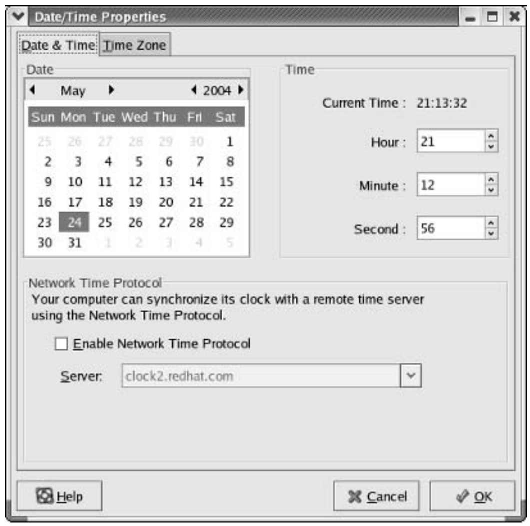
In the window shown in Figure 13.12, you can enable IP Forwarding, which lets your Linux computer work as a gateway between two or more networks. Changes you make are written to `/etc/sysctl.conf`.

Setting the Date and Time

Setting the right date and time for your computer can be important. If you're running an Internet store with servers in different time zones, you need to synchronize the time between the servers. Red Hat Enterprise Linux is configured to use the Network Time Protocol (NTP), which is part of the TCP/IP protocol stack.

You can set the date and time in `/etc/sysconfig/clock`. Alternatively, you can start the Red Hat Date/Time Properties tool by selecting Main Menu ➢ System Settings ➢ Date & Time, or you can run the `redhat-config-date` and `redhat-config-time` commands from a GUI command-line interface. This opens the Date/Time Properties window, shown in Figure 13.13.

FIGURE 13.13
Date/Time Properties tool



You can set the date and time yourself. Once you've accepted any changes, Linux changes the hardware clock on your computer. Alternatively, you can set your computer to synchronize its clock with a remote server. With NTP and a network connection, Linux can send a message to a central time server for the current date and time.

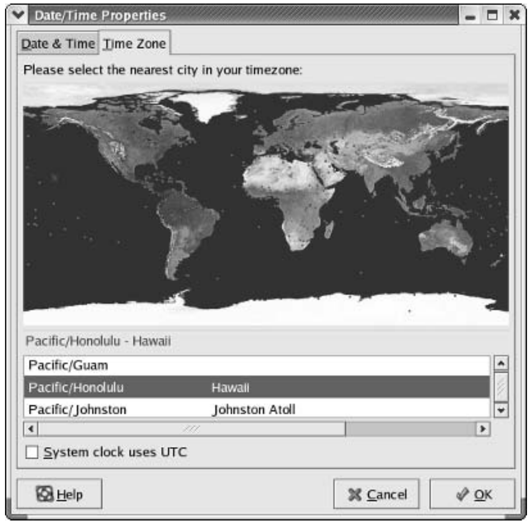
NOTE *If you give up control of your system clock to an NTP server, the Date/Time Properties window does not allow you to set the time independently.*

In any case, Red Hat Enterprise Linux also allows you to set the time zone associated with your computer. As you can see in Figure 13.14, you can set your computer to a wide range of time zones. The default is the standard U.S. East Coast time zone, listed as America/New_York. You can set your system to any one of several hundred locations.

Unless your computer is configured as a dual-boot with another operating system such as Microsoft Windows, you should activate the System Clock Uses UTC option. UTC is the French acronym for Coordinated Universal Time, which corresponds to Greenwich mean time. If you select UTC, Linux sets your hardware clock to this time and calculates the time zone difference to your location for the system clock.

Time zone changes are saved in `/etc/sysconfig/clock`; any NTP servers that you designate are recorded in `/etc/ntp/ntpservers`. Naturally, you can edit these files directly. Perhaps the most authoritative website on the NTP server is located at the University of Delaware at `www.eecis.udel.edu/~ntp`. It includes a link to a list of active NTP servers around the world.

FIGURE 13.14
Setting your time zone



The Red Hat Date/Time Properties tool automatically sets the NTP daemon, `ntpd`, to start the next time you boot Linux in runlevels 3 and 5. You can verify this with the following command:

```
# chkconfig --list ntpd
```

Summary

The `cron` daemon can help you run programs on an automated regular basis. Red Hat Enterprise Linux configures standard `cron` jobs through `/etc/crontab`, configured by time period in directories such as `/etc/cron.hourly` and `/etc/cron.weekly`. Users can configure their own `cron` jobs with the `crontab` command; each user's configuration is stored in the `/var/spool/cron` directory. `cron` security is governed by the `/etc/cron.allow` and `/etc/cron.deny` files.

The `at` command is like `cron`, except it can help you run jobs on a onetime basis. The `batch` command is a variation of `at` that runs a specified job when the demands on your system are less than 80 percent of capacity. Similar to `cron`, `at` command security is governed by the `/etc/at.allow` and `/etc/at.deny` files.

Another key to administering Linux is based on log files. Standard Linux log files are configured in `/etc/syslog.conf` and located in the `/var/log` directory. System logs help you trace detected hardware and analyze login activity. Daemon logs can help you monitor when daemons such as `crond`, `httpd`, and `smbd` are used. Other log files are available for tasks such as monitoring currently installed RPMs, secure connections, news servers, and more. You can even configure a group of servers to send their log files to a single computer.

Everyone who administers a Linux computer needs to know several basic process and user management commands. The `ps`, `top`, and `kill` commands help you find and kill processes that are out

of control. The `who` command can identify currently logged-in users. The `nice` and `renice` commands enable you to prioritize critical jobs.

There are a couple of other important tools available; the Kernel Tuning tool allows you to navigate the options for modifying how Linux interacts with your hardware. The Date/Time Properties tool allows you to set your computer to synchronize with a central NTP server on a regular basis.

In the following chapter, you'll extend your knowledge of Linux administration by learning the commands you need to back up all or part of your system.

Chapter 14

Backing Up Your System

DATA ON TODAY'S PERSONAL computers is fragile. Administrators are constantly worried about viruses, adware, and spyware that may affect data on a network. Attacks by crackers, power surges, mechanical failures, magnetic fields, and natural disasters can destroy some or all of the data on your hard drives.

The measures you take to back up your system depend on your situation. Backing up data for multiple users on multiple computers requires more care. To help recover from a disaster at your facility, you may choose to store data at a different site.

Several different types of backup media are available. You can back up critical data on CDs, or you can back up entire computers on portable or external hard drives or remote tape drives. Recordable DVDs are quickly becoming a viable alternative to tape drives. Alternatively, removable and external hard drives have the capacity and can easily be stored in remote locations. On larger networks, backups to a central server may be an option. As an administrator, you may want one location to back up files from all servers on your network. As a workstation or desktop user, you may find it convenient to have a central backup server maintained by a responsible Linux administrator.

Depending on your backup mode and media, other backup commands are also available to you, such as `tar`, `cpio`, `dump`, and `restore`. Alternatively, a properly configured Redundant Array of Independent—or Inexpensive—Disks (RAID) can also back up your data on other hard drives. In some cases, a RAID drive can be removed and stored in a secure remote location. This chapter covers the following topics:

- Exploring backup concepts
- Selecting your media
- Using backup and restore commands
- Understanding RAID

Exploring Backup Concepts

Selecting a backup strategy depends on the risks that you are willing to take. The risk equation for any computer backup consists of two parts. First, you need to understand what can happen to your

data and computers. Disasters range from a corrupted file to the destruction of your main corporate facility. Second, you need to select a backup strategy to address each of these disasters. The strategy (and cost) varies depending on the importance of the data, your users' reactions to different disasters, and how fast you need to restore from backup. Finally, you must make sure you can restore from any backup you create, before you really need it.

To understand these parts of the risk equation, you should examine various disaster scenarios and the available levels of data and computer backup.

Data Disaster Scenarios

The loss of even a single file can be a disaster for a user. The loss of a commercial airplane engineering drawing, a master's thesis, or even chapters for a book in production can be a life-changing event.

Information technology managers have to plan for every level of disaster, from the loss of a file to the effects of a nuclear war. (Yes, some corporate IT managers create backup plans for a nuclear war.) See Table 14.1 for several basic scenarios.

TABLE 14.1: DATA DISASTER SCENARIOS

Scenario	Recovery Strategy
Lost user file	Restore from backup of the `/home` filesystem.
Lost configuration file	Restore from backup of `/etc`.
Lost application file	Reload from backup, or reinstall the application.
Damaged partition	Restore partition from backup, or use an appropriate level of RAID.
Damaged hard drive	Restore hard drive from backup or an appropriate level of hardware RAID.
Damaged computer	Restore data from other computers or tapes/CDs/DVDs on site.
Damaged data facility	Restore from backups stored in a remote location.
Electromagnetic data loss	Restore from nonmagnetic backups.

This is far from a comprehensive list of possible disaster scenarios. For example, problems with a network can be just as difficult, especially if they prevent users from accessing their files or applications on a server. Of course, disaster planning for networks is beyond the scope of this book, but the principles are essentially the same.

Levels of Backup

You need to decide what data is critical to you. If you're a personal desktop user, you may have just a few critical files, such as documents. You may be able to back up these files every time you change them.

If you're a Linux administrator for a network of computers, you may be willing to spend a lot more money to protect and back up your data. However, with the amount of data stored in a network of computers, it may not be cost effective to back up everything every night.

In the enterprise, you could be working with both situations. While most of your computers are connected to your networks, plenty of people in remote locations could also need reliable large-scale

backups. And if you're administering a backup server, you could be storing that data on some external drive with multipath support.

NOTE *Multipath support provides more than one path between a computer and storage device (such as multiple Fibre Channel cables to SCSI drives). So if there's a failure in one cable, the other cable takes over, and the multipath software incorporated into Red Hat Enterprise Linux can automatically switch cables. It is most commonly associated with RAID, which we describe in the last part of this chapter.*

To some extent, with the use of commands such as `rsync`, and network drives that we'll cover in several future chapters, the basic commands for local and backups are essentially the same.

The following sections examine what you can do if you use a Linux computer as a personal desktop, administer a regular network, or administer a network where you have very time-sensitive data. What you actually do in practice may vary with the importance of the data and your available resources.

Your needs will also determine how often you do backups of time-sensitive data and the hard drives on a large group of computers.

Personal Desktop Users

Not all users back up their computers. Personal desktop users who just use their computers to browse the Internet may not have any irreplaceable data on their systems. For some home users, a disaster is just an inconvenience; all they need to do is reinstall their operating system and connect to the Internet once again. However, if you're a home user who keeps critical data such as financial records on your computer, consider yourself a Linux administrator and read the sections that follow.

In many cases, all these users need are backups of files on their home directories. Backups of configuration files in `/etc` can also help users restore many customized settings.

Some users prefer to back up all files and data on their Linux computers. That way, they can recover from any disaster without spending additional time reconfiguring their systems.

Linux Administrators

If you're the Linux administrator responsible for a group of computers, timely backups are critical. For example, the data associated with the design of a new airplane evolves constantly.

Though it may not be too difficult to recover data from a lost day of work, the consequences of a lost week or month of design work for an airplane company can be rather expensive. In this case, you can configure a series of nightly backups on larger capacity media, such as DVDs or tape drives. Such drives can be organized in large groups, known colloquially as *jukeboxes*. With such hardware, you can back up a substantial amount of data at relatively high speeds to hundreds of DVDs or tape drives.

In this way, a Linux administrator can help tired engineers recover the data they accidentally deleted. If there's a larger disaster, the administrator can reinstall Linux, along with the appropriate engineering software, and then restore the design files to the appropriate directories.

Time-Sensitive Situations

Computers are used in time-sensitive situations. For example, if you're the Linux administrator responsible for a financial services firm, timely backups are critical. For example, if you are unable to

restore the data associated with sales in the stock market, the consequences can be expensive. Time-sensitive information suggests the need for real-time backups, such as those associated with RAID.

In this way, the failure of any hard disk doesn't affect the operation of the firm. With the use of removable hard disks, RAID data can also be copied and stored in external locations.

Backup Type and Frequency

The most straightforward backup is of everything on your computer. However, as the amount of data on individual hard disks moves into the hundreds of gigabytes, the amount of time required can stretch into dozens of hours.

Although Linux computers are multitasking, the load associated with a backup can affect performance for your users. That leaves you with two basic choices: back up your entire computer only on occasion (for example, weekends) or back up only part of your data, such new files or the `/home` and `/etc` directories. This is one area where the `rsync` command can help. As we describe later, it backs up only the parts of each file that have changed.

Many Linux administrators use a mix of the two philosophies—a complete backup available for a Linux computer, with daily backups for new files. There are two ways to make this happen:

Differential backup A differential backup includes all files that were created or changed since the last full backup. As time increases since the last full backup, the size of a differential backup gets progressively larger. Restoring a system requires only the data you saved in the full, and the differential backup.

Incremental backup An incremental backup includes all files that were changed since the last backup of any type. Incremental backups are almost always smaller than differential backups. However, restoring a system from an incremental backup can be more difficult. It requires the data you saved in the full backup, the differential backup (if applicable), and all of the subsequent incremental backups.

Because of the time associated with restoring data, many Linux administrators use some form of RAID. As you'll see later in the chapter, RAID can provide approximate real-time redundancy for your data.

Selecting Your Media

You can back up data anywhere you can record information. In some cases, you may even want to print hard copies of key configuration files. Some personal desktop users may find 1.44MB floppy drives adequate. Workstation users may find that slightly larger capacity media such as 100MB Zip or 230MB Bernoulli drives meet their needs. In either case, users can back up just the critical files they need, usually from their `/home` directory. Commands such as `tar` and `cpio` let you back up specific groups of files and or directories, as described later in this chapter.

For those with a need to back up gigabytes (GB) or even terabytes (TB) of data (1TB = 1000GB), there are three basic options: tape drives, writeable CDs/DVDs, and removable/external hard disks. These options can be either directly connected to the computer or connected to a backup server via a network. If one tape or CD is not enough to back up the user data (specifically, the `/home` directory) on your hard disk, hardware is available that organizes these systems into tape libraries and CD/DVD

jukeboxes. One way to use hard disks for backups is discussed later in the chapter in the section, "Understanding RAID."

You can copy and transport all three types of media to secure and remote locations. If your facility is destroyed by fire or some other disaster, the right media, properly stored, can help you restart your business or organization. Backups were tested on a large scale during the tragedies of September 11, 2001. Some financial institutions saved data in remote locations in real time; other businesses were able to get to their data in hours or days.

Note *A number of other third-party software solutions are available; you may need their support if you have the amount of data that justifies a jukebox or a high-capacity tape drive. You'll find a list of third-party backup software and hardware manufacturers at* `www.storagesearch.com`.

Tape Drives

If you have the budget, you can get a tape drive that can store your data nearly as fast as current IEEE 1394 and USB 2.0 hard drives. As of this writing, systems are available that can store nearly 30TB of data in over 100 tape cartridges in a single box. With data transfer speeds of nearly 1,000 GB per hour, it is possible to fill this unit with a full backup in a single weekend. In fact, data transfer to these drives is often faster than to many conventional internal hard drives.

Also available are lower-capacity, less-expensive tape drives with conventional interfaces, such as to parallel ports, IDE, and SCSI. Tape drives with these interfaces carry device names similar to hard drives with these connections. Generally, drives with parallel port connections are far too slow for current standard hard drives. External IDE or SCSI hard drives offer speeds similar to internal drives with the same type of connection.

Also, two tape drives have USB and IEEE 1394 interfaces. As discussed in Chapter 2, support for IEEE 1394 and many USB interfaces is still officially experimental, and they may not work with Red Hat Enterprise Linux. However, many work well. I have a couple of portable hard drives connected via an IEEE 1394 connection, and the effective data transfer speed is actually faster than to the internal hard drive.

If you're backing up to a tape drive, you may consider installing AMANDA, the Advanced Maryland Automatic Network Disk Archiver. It allows you to back up files and directories from multiple computers to a single tape drive connected to your network. For more information, read the online chapter on this utility at `www.backupcentral.com/amanda.html`. Unfortunately, as of this writing, AMANDA does not support backups to anything but a tape drive.

CD/DVD Backups

Compared to tape drives, writeable CDs and even DVDs seem to pale by comparison. A CD can hold only about 650MB of data; various DVDs can hold 4.7–17GB of data. But a number of jukeboxes are available that can write data to hundreds of disks.

In addition, CDs and DVDs hold a number of advantages over tape drives. In proper environmental conditions (in other words, don't store your CDs in a hot, humid environment!), CDs and DVDs can last for a decade or more. Unlike with tape drives or hard disks, you can't accidentally erase them with a magnet. (Remember, power tools can give off magnetic fields, which can make tape drives or even external hard drives problematic in an industrial setting.) They are not susceptible to the electromagnetic pulses associated with nuclear explosions.

Using Backup and Restore Commands

The commands you use depend in part on how you're backing up your data. Generic backups commonly use the `tar` or `cpio` commands. Alternatively, you can `dump` and `restore` data to and from a tape drive. If you're connected to a shared network directory, it doesn't matter whether the backup is to a local hard drive or to a remote directory connected over the network.

Backups to local CDs are associated with the `mkisofs`, `cdrecord`, and `dvdrecord` commands. Some variations are required to back up and restore data through the network to remote locations.

Generic Backup Commands

Let's look at the two generic Linux commands for backing up a group of files. The `tar` command was originally developed to archive files and directories to tape drives; the `cpio` command also copies files and directories to and from an archive. With the right options, these commands can be used to back up files to most media.

NOTE *You can also use the* `dd` *command to dump the contents of a directory directly to a device—for example, a floppy drive device such as* `/dev/fd0` *or a tape drive device such as* `/dev/st0`*. For an example on how* `dd` *is used, see Chapter 3.*

Archiving by *tar*

You examined the `tar` command for the first time in Chapter 10. It's simple to use. The format is easily compressed and downloadable. This command is the main alternative to the RPM system for packaging programs and applications. With the right options, it's functionally similar to the `.zip` file system associated with Microsoft Windows.

The `tar` command is designed to copy a series of files into a single large file. If you want to back up the files in mj's home directory, you can run the following command:

```
# tar cvzf mjbackup.tar.gz /home/mj
```

This command creates (`c`) a backup, listing every filename in the archive (`v` = verbose) in compressed format (`z` = zip) in the file (`f`) named `mjbackup.tar.gz`. Files in subdirectories of `/home/mj` are also saved to this archive. You can then save this archived file to a backup area such as a network share or a tape drive.

NOTE *Compressed tar archives often include the* `.tar.gz`*,* `.tgz`*, and* `.tar.bz2` *extensions. The first two extensions are both tar archives compressed with the* `gzip` *command. The last extension, based on the bzip2 "Burrows-Wheeler block sorting compression algorithm," is slightly more efficient at data compression.*

You can just as easily unarchive files with the following command:

```
# tar tkvzf mjbackup.tar.gz
```

This command lists (`t`) the files in your archive. When it restores, it does not overwrite your current files (`k` = keep old files). In verbose (`v`) mode, you see everything that happens. If you stored files in a zipped format, you need to restore from the zipped (`z`) format. Also, it is restoring from the backup file named `mjbackup.tar.gz`.

You can review some of the available `tar` switches in Table 14.2. Note that the first switch in the `tar` command should start with a `c`, a `t`, or an `x`.

NOTE *The* `tar` *command is path dependent. If you save the files in a directory using the absolute path (with a leading forward slash, such as* `/home/mj`*), you can restore the files to that directory from any location on that computer. Alternatively, if you use the relative path (without a leading forward slash, such as* `home/mj`*), files may not be restored to their original locations; it depends on the present working directory.*

You can use a number of `tar` commands to create and extract archives. Some typical commands include the following. Read them over using the descriptions in Table 14.2.

```
# tar xzvf download.tar.gz
# tar czvf backup.tar.gz /somedirectory
```

TABLE 14.2: COMMAND OPTIONS FOR *TAR*

OPTION	FUNCTION
c	Creates an archive
d	Compares files between an archive and a current directory
f	Uses the following filename for the archive
j	Compresses in bzip2 format to or from an archive
k	Does not overwrite existing files
r	Adds files to the end of an archive
t	Lists files in a current archive
v	Verbose; lists all files going in or coming out of an archive
z	Zip; compresses files to or from an archive in regular gzip format

NOTE *The* `tar` *command is similar to* `ps` *in that single-letter command options do not require a leading dash.*

ARCHIVING BY *CPIO*

The `cpio` command can help you archive a class of files, because unlike `tar`, it works with standard input and output. This use is suggested by its name (`cpio` = copy + input/output).

As with `tar`, it's fairly easy to archive known directories (along with the files in their subdirectories). For example, if you want to back up the files in mj's home directory, you run the following command:

```
# find /home/mj | cpio -o > mjarch.cpio
```

But this has a disadvantage; `cpio` takes from standard input and archives to standard output. Note how the standard input, all files in the `/home/mj` directory, is piped to the `cpio` command. Since this

works with classes of files, you can use wildcards to set up a group of files as standard input as well. For example, the following command creates an archive from the `.tif` files in the current directory:

```
# find *.tif | cpio -o > mjtifs.cpio
```

Remember, the `find` command is flexible; the following command creates an archive from all the `.tif` files on your system:

```
# find / -name '*.tif' | cpio -o > mjtifs.cpio
```

It's easy to restore the files from a `.cpio` archive. The following command restores the files in the `mjarch.cpio`:

```
# cpio -i < mjarch.cpio
```

As with `tar`, the way `cpio` restores files saved from a directory depends on whether you used the absolute or relative path.

One of the advantages of `cpio` is the ability to send files directly to external sources. For example, the following commands send and restore the files from mj's home directory to a SCSI tape drive:

```
# find /home/mj | cpio -o > /dev/st0
# cpio -i < /dev/st0
```

A number of options are available for the `cpio` command. Table 14.3 describes some of the important options.

Table 14.3: ***cpio*** **Command Options**

Option	Function
`-A`	Appends to an existing archive; closely associated with `-F`
`-F`	Specifies archive filename; can substitute for redirection arrow (>)
`-i`	Extracts from an archive file or device
`-o`	Copies to an archive file or device
`-u`	Replaces all files, even if they're newer
`-v`	Verbose mode

Tape *dump* and *restore*

The `dump` and `restore` commands make it easy to implement incremental and differential backups. `dump` allows you to take the contents of a directory, and `restore` allows you to interactively return backed-up files to their original locations.

Although these commands are most commonly associated with tape drives, they work with other media as well. The examples shown in the following sections are based on using these commands to back up a home directory to a floppy disk.

ARCHIVING BY *DUMP*

The `dump` command has three basic levels of options. You can set up a series of commands that starts with a full backup of a home directory, followed by differential backups. For example, if you want to back up the home directory of mao with `dump` to the `/dev/nst0` tape drive, you run the following commands:

```
# dump 0f /dev/nst0 /home/mao
# dump 1f /dev/nst0 /home/mao
# dump 2f /dev/nst0 /home/mao
# dump 3f /dev/nst0 /home/mao
# dump 4f /dev/nst0 /home/mao
# dump 5f /dev/nst0 /home/mao
```

The first command, with the `0f` option, sets up a full backup of the `/home/mao` directory. The commands that follow, when run in sequence, set up incremental backups that save only those files that were changed since the previous backup.

TIP *To speed the backup, you may be able to use the biggest block size allowed by your backup system (for instance, a tape drive). For example, the command* `dump 0f /dev/nst0 /home/mao -b 2048` *uses a block size of 2,048 bytes. You may want to experiment with larger block sizes to reduce backup time. But remember, you should also verify the results of your experiment with the appropriate* `restore` *command.*

Alternatively, you could start with a full backup, followed by differential backups with a sequence of commands, such as the following:

```
# dump 0f /dev/nst0 /home/mao
# dump 8f /dev/nst0 /home/mao
# dump 7f /dev/nst0 /home/mao
# dump 6f /dev/nst0 /home/mao
# dump 5f /dev/nst0 /home/mao
# dump 4f /dev/nst0 /home/mao
```

NOTE *You don't need to run all six of these commands. With a differential backup, you just need to make sure that the next number, such as* `4f`*, is lower than the previous differential backup command. Otherwise, the backup may not get properly recorded.*

If you're backing up an entire filesystem, you'll want to use the `u` option, which stores the history in `/etc/dumpdates`. For example, the following command backs up the entire root (/) directory filesystem:

```
# dump 0uf /dev/nst0 /
```

Take a look at the workings of a `dump` command on the files in the `/home/mao` directory in Figure 14.1.

FIGURE 14.1
dump output

```
[root@Enterprise3 root]# dump 0f /dev/fd0 /home/mao/
  DUMP: Date of this level 0 dump: Fri Mar 26 12:04:07 2004
  DUMP: Dumping /dev/hdd1 (/home (dir /mao)) to /dev/fd0
  DUMP: Added inode 8 to exclude list (journal inode)
  DUMP: Added inode 7 to exclude list (resize inode)
  DUMP: Label: none
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 81 tape blocks.
  DUMP: Volume 1 started with block 1 at: Fri Mar 26 12:04:07 2004
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /dev/fd0
  DUMP: Volume 1 completed at: Fri Mar 26 12:04:08 2004
  DUMP: Volume 1 90 tape blocks (0.09MB)
  DUMP: Volume 1 took 0:00:01
  DUMP: Volume 1 transfer rate: 90 kB/s
  DUMP: 90 tape blocks (0.09MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 0 dump: Fri Mar 26 12:04:07 2004
  DUMP: Date this dump completed:  Fri Mar 26 12:04:08 2004
  DUMP: Average transfer rate: 90 kB/s
  DUMP: DUMP IS DONE
[root@Enterprise3 root]#
```

There are a number of options available for the `dump` command. Table 14.4 shows some of the important options.

TABLE 14.4: *DUMP* COMMAND OPTIONS

OPTION	FUNCTION
`0-9`	Dump level. 0 = full backup. Incremental backups use `dump` with increasing numbers (for example, 1, 2, 3…). Differential backups use `dump` with decreasing numbers (for example, 8, 7, 6…).
`A`	Archives a table of contents for the backup.
`f`	Writes the backup to a file or device.
`j` *`level`*	Writes with compression; you need to specify a compression level such as 2 or 4.
`T`*`date`*	Uses the specified date instead of what's shown in `/etc/dumpdates`.
`u`	Updates `/etc/dumpdates` after a successful backup.

RECOVERING WITH *RESTORE*

You have two ways to restore from a backup created with the `dump` command: interactively or directly. In either case, you can restore an entire backup or just the files you need.

You can view a listing of files that were backed up with the `dump` command. As shown in Figure 14.2, the following command lists the files from the backup of mao's home directory:

```
# restore -tf /dev/fd0
```

Alternatively, you can use restore mode to search through a current backup. As shown in Figure 14.3, the `-i` option brings you into interactive mode, where you can use some basic Linux navigational commands.

FIGURE 14.2
Files on a backup

```
[root@Enterprise3 root]# restore tf /dev/fd0
Dump   date: Fri Mar 26 12:04:07 2004
Dumped from: the epoch
Level 0 dump of /home (dir /mao) on Enterprise3:/dev/hdd1
Label: none
         2      .
     49172      ./mao
     49173      ./mao/.kde
     49174      ./mao/.kde/Autostart
     49175      ./mao/.kde/Autostart/.directory
     49176      ./mao/.emacs
     49177      ./mao/.bash_logout
     49178      ./mao/.bash_profile
     49179      ./mao/.bashrc
     49180      ./mao/.gtkrc
     49181      ./mao/.zshrc
[root@Enterprise3 root]#
```

FIGURE 14.3
An interactive restore

```
[root@Enterprise3 root]# restore -if /dev/fd0
restore > ls
.:
mao/

restore > cd mao
restore > ls
./mao:
.bash_logout   .bashrc        .gtkrc         .zshrc
.bash_profile  .emacs         .kde/

restore > help
Available commands are:
        ls [arg] - list directory
        cd arg - change directory
        pwd - print current directory
        add [arg] - add `arg' to list of files to be extracted
        delete [arg] - delete `arg' from list of files to be extracted
        extract - extract requested files
        setmodes - set modes of requested directories
        quit - immediately exit program
        what - list dump header information
        verbose - toggle verbose flag (useful with ``ls'')
        prompt - toggle the prompt display
        help or `?' - print this list
If no `arg' is supplied, the current directory is used
restore >
```

A number of options are available for the `restore` command. Table 14.5 lists some of the important ones.

TABLE 14.5: ***RESTORE*** **COMMAND OPTIONS**

OPTION	FUNCTION
`-C`	Compares a backup with current files.
`-f`	Specifies a file.
`-i`	Allows interactive recovery from a backup; several commands are available in `restore` mode.
`-r`	Rebuilds the data to a freshly formatted partition.

Continued on next page

Table 14.5: ***restore*** **Command Options** *(continued)*

Option	Function
`-t`	Lists the filenames in the backup.
`-x`	Extract the files to the current directory.

As you can see in Figure 14.3, the files in the floppy backup are in the `mao/` subdirectory. Thus, before you can restore files to `/home/mao`, you'll need to navigate to the `/home` directory. Therefore, if user mao complains that his files are deleted, you'd put the floppy in the drive and run the following commands:

```
# cd /home
# restore -xf /dev/fd0
```

Backup Commands for CDs/DVDs

Before you can start recording data, you need to check whether Red Hat Enterprise Linux recognizes your hardware. This is normally a simple exercise. Then you can create files suitable for CDs or DVDs and record them with the appropriate commands.

Checking Hardware

Before you can start backing up data to your writeable CD or DVD drive, you need to make sure it's actually working. Assuming Red Hat Enterprise Linux has automatically detected the right drive, you should see the appropriate setting for it when you issue one of these commands:

```
# cdrecord -scanbus
# dvdrecord -scanbus
```

Linux uses SCSI drives for recording. But that's probably not a big deal if all you have is an IDE drive; in most cases, Red Hat Enterprise Linux automatically configures SCSI emulation by default. In other words, it makes your IDE CD or DVD writer look like a SCSI drive.

From either of these commands, you should see output associated with a `scsibus`, similar to the following. Though this component is listed as a CD-ROM, it works as it should as a DVD-RW drive:

```
0,0,0  0) 'DVD-RW   '  'IDE1004   ' '0043' Removable CD-ROM
```

However, if you get a message such as "No such file or directory" or "cannot open SCSI driver," there's a problem. Red Hat may be having a bit of trouble adapting your system to SCSI emulation. To be sure, check your `/proc/scsi/scsi` file. If there are no SCSI devices (including emulated SCSI devices) on your system, this file will be empty. If `/proc/scsi/scsi` is empty, check your `dmesg` messages. Make sure Linux detects your CD or DVD drive. If the drive isn't detected, you may have a hardware problem. The Enterprise Linux kernel is configured to detect almost all IDE and SCSI drives; however, support for old CD drives is disabled by default.

It's not hard to set up SCSI emulation. In your bootloader, all you need to do is add a kernel command. For example, in the GRUB bootloader, you may find a kernel command such as the following:

```
kernel /vmlinuz-2.4.21-9.EL ro root=LABEL=/ hdd=ide-scsi
```

The `hdd` in this command corresponds to the drive as detected and documented through `dmesg` command output. The `ide-scsi` command sets up SCSI emulation of this IDE CD or DVD drive.

In most cases, Red Hat Enterprise Linux automatically detects these drives and adds the appropriate `ide-scsi` command to your bootloader. However, if you do something such as add your own DVD drive, you may need to add the command yourself.

Making an Image

The next step in setting up files to write to a CD or DVD is to make an image file. Whether you're recording to a CD or a DVD, you can create the image file with the `mkisofs` command. As an example, assume you want to back up all the files and directories under `/home`. You can use the following command, where `-r` includes Rock Ridge extensions (which supports Unix-based filesystems), `-J` includes the Joliet filesystem (which makes files readable under Microsoft operating systems), `-T` preserves long filenames, and `-o` stands for output:

```
# mkisofs -J -r -T -o newcd.iso /home
```

This may create a very big file; if you're creating an image for a DVD, the file could easily be several gigabytes in size. It's a good idea to check the integrity of this file. One way to do this is to mount the image file as if it were a CD or DVD. For example, the following command mounts your newly created `newcd.iso` image on `/mnt/cdrom`:

```
# mount -t iso9660 -o loop newcd.iso /mnt/cdrom
```

Alternatively, the following command lists the files in the appropriate image file:

```
# isoinfo -i newdvd.iso -l
```

Both commands work on any sort of ISO file.

Burning the Image

Now we're ready to copy the image to a blank writeable CD. The `cdrecord` command can help. For the items cited in the previous section, you'd use this command:

```
# cdrecord -v speed=4 dev=0,0,0 newcd.iso
```

The `-v` option allows you to see what happens as Linux copies the image onto your CD. If there is a problem, these messages can also help you diagnose the cause. Figure 14.4 shows how to create a Red Hat installation boot CD from the `boot.iso` file described in Chapter 4. As you can see in the figure, a substantial number of useful messages are available when you run this command.Unfortunately, in its current form, it can't handle more data than on a regular CD, so it isn't useful for DVDs.

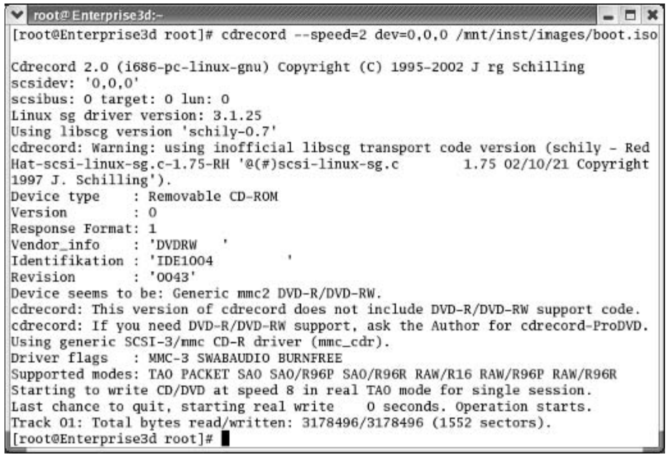


FIGURE 14.4
The cdrecord process

NOTE *If you're in GNOME and insert a blank writeable CD, Nautilus automatically opens a* `burn:///` *window, where you can copy the files and folders that you want written to that CD. It includes a Write To CD button and easy to understand prompts.*

BURNING A DVD IMAGE

There are three basic standards for DVD recorders: DVD-RAM, DVD-R/-RW, and DVD+R/+RW. Generally, if you record a DVD using one standard, you can't play that DVD on a drive of a different standard. However, most current DVD recorders and data readers work with multiple standards.

However, interchangeability is far from complete. There are two basic RPM packages that you can use with DVDs: `dvdrecord` and `dvd+rw-tools`. These RPMs (as well as `cdrecord`) are installed as part of the Sound and Video package group. Let's take an example using both RPMs.

First, I wanted to set up all of the installation files on a single DVD. To configure an ISO from the 2GB of installation files stored on the `/mnt/inst` directory, I've run the following command:

```
# mkisofs -J -r -T -o newcd.iso /mnt/inst
```

Second, if I have a DVD-RAM writer, I can write the files from the ISO image to a blank DVD-RAM with the following `dvdrecord` command:

```
# dvdrecord -v speed=1 -dao dev=0,1,0 newdvd.iso
```

This records the `newdvd.iso` image, verbosely (`-v`), at first speed (`speed = 1`), in Disk at Once (`-dao`) mode, where data is written in a single operation.

Alternatively, for another kind of writer such as DVD+RW or DVD-RW, you'll use the commands from the `dvd+rw-tools` RPM. One advantage is you don't need a separate ISO file.

Now you'll need to format the DVD disk. You can do this with the `dvd+rw-format` command. You'll need to know the device file associated with your DVD; normally, it's linked to `/dev/cdrom`, which you can check with the following command:

```
# ls -l /dev/cdrom
lrwxrwxrwx  1 root root  9 Mar 26 18:55 /dev/cdrom -> /dev/scd0
```

Now you can format your new DVD with the following command:

```
# dvd+rw-format /dev/scd0
```

This takes a few minutes. Once format is complete, you can start the recording process with the `growisofs` command:

```
# growisofs -Z /dev/scd0 -R -J /mnt/inst
```

Working backward in this command, it takes the files in the `/mnt/inst` directory, with Joliet (`-J`) and Rock Ridge (`-R`) extensions. It writes the files to the device associated with `/dev/scd0`, and it's the first session (`-Z`) on this DVD.

Red Hat Enterprise Linux 3 does not include a man page for this command; other useful options include (`-M`) for second (and later) sessions; and (`-speed=n`) to regulate the write speed.

NOTE *Commands for recording data on DVDs are still under development; thus the commands shown in this chapter are subject to change. For the latest information, see the official website of the* `dvdrtools` *project at* `www.nongnu.org/dvdrtools`.

Transferring Fast with *rsync*

A quick way to transfer files between directories is to use the `rsync` command. Its strength is after the first full backup; the only data that is sent between directories are the parts of each file that have changed. You can even make the transfer using secure services.

If you've never used this command before, start with a basic command. For example, you can copy the contents of a mounted CD (on `/mnt/cdrom`) to the local `/var/ftp/pub/inst` directory with the following command:

```
# rsync -a /mnt/cdrom/* /var/ftp/pub/inst
```

So far, this looks like a simple copy command. But the power of `rsync` comes with its use over a network. But before you can use it on a network, you'll need to set it up as a server. It's an `xinetd` server, which I describe in detail in Chapter 18. For now, you can start the `rsync` service (and make sure it starts the next time your computer boots) with the following command:

```
# chkconfig rsync on
```

Its origins in the Remote Shell (`rsh`) commands are problematic, as that's not a secure service. However, you can set it up to use the Secure Shell (`ssh`) service, which I also describe in Chapter 18.

Now look at the previous `rsync` command again. What if you wanted to set it up on an FTP server on a different computer? Well, assume that you have a computer named ftpserver on your network. (As usual, you can substitute the IP address). You'd use the following command to copy the contents of the local CD to the remote FTP server directory:

```
# rsync -av -e ssh /mnt/cdrom/* ftpserver:/var/ftp/pub/inst
```

This command copies all files in verbose mode (`-av`) from the source directory (`/mnt/cdrom/*`). It runs over a Secure Shell (`-e ssh`) using your current username. It copies the files to the remote computer named ftpserver, on that computer's `/var/ftp/pub/inst` directory. If you wanted to use a different user name such as donna, the command would change slightly.

```
# rsync -av -e ssh /mnt/cdrom/* donna@ftpserver:/var/ftp/pub/inst
```

Understanding RAID

Two different labels are associated with RAID: Redundant Array of Independent Disks and Redundant Array of Inexpensive Disks. Neither works in this case, because they don't accurately describe how the software version of RAID works in Red Hat Enterprise Linux.

A Redundant Array of Independent Disks implies that every disk in a RAID array is physically independent. If one disk fails, the others in the array can take over its functionality. Several versions of RAID exist where Linux can use the other working disk drives to reconstruct the data on any single failed disk drive. One version of RAID includes two separate hard disks with identical information.

You can also include "spare" hard disks in a RAID array. If there is a failure in any RAID 1 or RAID 5 hard disk, Linux can immediately begin rebuilding the data on the spare disk.

Using a RAID array provides three main advantages.

High availability A RAID array always lets you get to your data. With appropriate hardware, you can even change a hard disk while the computer is on. This isn't possible with a Red Hat Enterprise Linux software RAID array.

Fault tolerance With most hardware RAID arrays, the data is always accessible even if one hard disk fails. You can set up fault tolerance in a Red Hat Enterprise Linux software RAID array, as long as you configure each RAID partition in the array on separate physical hard drives.

Failover When a hard disk fails, a RAID system can automatically switch to a reserve hard disk. Data is automatically transferred to the backup hard disk or partition.

RAID Options

Three versions of RAID are associated with Red Hat Enterprise Linux: RAID 0, RAID 1, and RAID 5. Briefly, RAID 0 can speed access to hard disks, without fault tolerance. RAID 1, since it has two separate disks with identical information, complete fault tolerance. RAID 5 is based on an array of three or more disks and also provides fault tolerance.

You learned to configure a basic RAID array during the Red Hat Enterprise Linux installation process, as discussed in Chapter 3. You can also configure or revise your RAID configuration, as described in this chapter. Red Hat provides a number of sample RAID configuration files in the `/usr/share/doc/raidtools-1.00.3` directory. The filenames, such as `raid1.conf.sample`, are straightforward. You can modify them for your configuration and save or append them to your `/etc/raidtab` configuration file.

NOTE *Although the Linux kernel also supports RAID 4, the Red Hat Enterprise Linux installation program doesn't support configuring this version of RAID. There is one difference between RAID 4 and RAID 5. In RAID 4, all parity information is stored on one partition or hard disk. In RAID 5, parity information is distributed on all hard disks in the array.*

Configuring RAID 0

This level of RAID includes two or more drives or partitions, grouped together. When these are separate physical drives, your computer can use the buffers on each drive. This is one way RAID 0 can speed reading and writing to your hard disks.

However, RAID 0 provides no data redundancy. In other words, if any disk or partition in a RAID 0 array fails, you lose all of the data in that array.

NOTE *RAID 0 is sometimes known as "striping without parity."*

Configuring RAID 1

At this level, RAID is like a mirror. It includes two separate disks or partitions with identical data. When RAID 1 is used for two separate hard disks, either hard disk can be used. If one hard disk fails, the other hard disk is ready to step in. No data is lost.

The drawback to RAID 1 is that it takes longer to write data to disk. With RAID 1, writes are not complete until the data is written to both disks. The hardware version of RAID 1 is secure but expensive; if you were to implement RAID 1 on all of your computers, you would need to purchase and install twice as many hard disks.

NOTE *RAID 1 is sometimes known as "disk mirroring."*

Configuring RAID 5

At this level, RAID requires three or more disks. RAID 5 stripes parity information evenly across all disks in the array. If one disk fails, Linux can reconstruct the "lost" information from the parity data on the remaining disks. Although data retrieval is slower when a RAID 5 disk fails, your system can still run.

If a spare hard disk is available in a RAID 5 array, Linux immediately begins to write this lost information onto the spare disk.

This level of RAID is generally preferred in most cases. Data integrity is ensured. The space of only one disk is sacrificed to hold the parity information. And performance is good.

NOTE *RAID 5 is sometimes known as "disk striping with parity."*

Software and Hardware RAID

The software RAID that you can configure in Red Hat Enterprise Linux is a bit different from the hardware RAID, because it uses partitions, not separate physical disk drives. If you use RAID on Red Hat Enterprise Linux, I highly recommend you avoid using partitions from the same hard disk for any single RAID array. Otherwise, any failure of that hard disk could destroy all data in that RAID array.

Several hardware RAID systems are available, with their own software support for Linux. However, the principles discussed in this chapter work for any version of RAID associated with Red Hat Enterprise Linux.

Dedicated RAID hardware can help ensure that your data survives any catastrophic physical failure on any single hard disk.

Creating RAID Partitions

You can create RAID partitions after installing Red Hat Enterprise Linux. As an example, assume you have several SCSI hard disks available. You've installed Red Hat Enterprise Linux on the first SCSI hard disk, `/dev/sda`. You have three other hard disks available for a RAID array, `/dev/sdb`, `/dev/sdc`, and `/dev/sdd`.

NOTE *You can also create RAID partitions during the Red Hat Enterprise Linux installation process. See Chapters 3 or 4 for details.*

After installation, the standard utility for creating new partitions is `fdisk`. For more information on the basics of this utility, please refer to Chapter 7.

To create a RAID array in Red Hat Enterprise Linux, you need two or more partitions of approximately equal size. If you want your array to survive the failure of any physical hard drive, each of the partitions in a RAID array must be on a separate physical drive.

Once you have the partitions for a RAID array, you can use the `fdisk` utility to change the partition type to one suitable for a RAID array. For example, the commands shown in Figure 14.5 change the partition `/dev/sdb1` to one that you can make part of a RAID array.

NOTE *Never change the file type of a partition with data you need, unless you've already backed it up in a secure location. When you use* `fdisk` *to change the file type, that action can destroy any data currently stored on that partition.*

Once you've created the disks or partitions for your RAID array, you'll need to format them. As discussed in Chapter 7, you need the `mkfs -j` *`partitiondevice`* command to format your new partition to the ext3 filesystem. For example, the following command properly formats the partition just created:

```
# mkfs -j /dev/sdb1
```

Repeat the process to create the RAID partitions you need. Remember to format all the partitions that you're using in your RAID array.

FIGURE 14.5
Creating a RAID partition

```
[root@Enterprise3 root]# fdisk /dev/sdb

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes
128 heads, 32 sectors/track, 512 cylinders
Units = cylinders of 4096 * 512 = 2097152 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1             1       200    409584   83  Linux
/dev/sdb2           201       300    204800   83  Linux
/dev/sdb3           301       350    102400   83  Linux

Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@Enterprise3 root]#
```

Configuring */etc/raidtab*

When you have the RAID partitions you need, the next step is to edit the RAID configuration file, `/etc/raidtab`. This file is fairly easy to configure. The following sections illustrate example configurations for RAID 0, RAID 1, and RAID 5 arrays.

If you're not sure where to start, a number of sample files are available in the `/usr/share/doc/raidtools-1.00.3` directory. Their names are straightforward; for example, the `raid5.conf.sample` file demonstrates a sample `raidtab` configuration file for a RAID 5 array.

You can use several commands in `/etc/raidtab` for any of these arrays; some of the more important commands are shown in Table 14.6.

TABLE 14.6: COMMANDS IN */ETC/RAIDTAB*

COMMAND	FUNCTION
`raiddev`	RAID device filename.
`raid-level`	RAID array type, usually 0, 1, or 5.
`nr-raid-disks`	Number of disks assigned to this RAID array.
`nr-spare-disks`	Number of backup disks assigned to this RAID array.
`persistent-superblock`	If this =1, Linux can detect and automatically start this RAID array.
`chunk-size`	Amount of data to be read/write, in KB.
`parity-algorithm`	How RAID 5 calculates parity.
`device`	Device name of a RAID partition.
`raid-disk`	Number assigned to a partition in a RAID array, in sequence, starting with 0.
`spare-disk`	Number assigned to a reserve partition in a RAID array, in sequence, starting with 0.

RAID 0 */ETC/RAIDTAB*

RAID 0 is disk striping without parity. Because there is no data redundancy, no spare disk partition is configured in this RAID array. The following excerpt from a RAID 0 `/etc/raidtab` file configures a RAID array of two partitions, `/dev/sda1` and `/dev/sdb1`, with a fairly large chunk-size (16KB), to maximize data transfer speed:

```
raiddev     /dev/md0
raid-level  0
persistent-superblock    1

nr-raid-disks   2
nr-spare-disks  0
chunk-size  16

device     /dev/sda1
raid-disk  0
device     /dev/sdb1
raid-disk  1
```

RAID 1 */ETC/RAIDTAB*

RAID 1 is known as disk mirroring. Because this is the ultimate in redundancy, one spare disk partition is included in the following excerpt from `/etc/raidtab`. The two partitions in the array are `/dev/sda2` and `/dev/sdb2`. The spare partition is `/dev/sdc2`.

```
raiddev     /dev/md1
raid-level  1
persistent-superblock    1

nr-raid-disks   2
nr-spare-disks  1
chunk-size  4

device     /dev/sda2
raid-disk  0
device     /dev/sdb2
raid-disk  1
device     /dev/sdc2
spare-disk 0
```

RAID 5 */ETC/RAIDTAB*

RAID 5 is known as striping with parity. This can be run with a large number of disks or partitions. Since it provides redundancy, two spare disk partitions are included in the following excerpt from `/etc/raidtab`. The four RAID partitions in the array are `/dev/sda3`, `/dev/sdb3`, `/dev/sdc3`, and `/dev/sdd3`. The spare partitions are `/dev/sde3` and `/dev/sdf3`.

```
raiddev     /dev/md2
raid-level  5
persistent-superblock    1

nr-raid-disks   4
nr-spare-disks  2
chunk-size  4

device     /dev/sda3
raid-disk  0
device     /dev/sdb3
raid-disk  1
device     /dev/sdc3
raid-disk  2
device     /dev/sdd3
raid-disk  3
device     /dev/sde3
spare-disk 0
device     /dev/sdf3
spare-disk 1
```

Creating the RAID Device

OK, we're almost there! You've created the partitions you want in your RAID array. You've set them to the Linux RAID file type. You've formatted each partition. You've set up the configuration for the RAID array in `/etc/raidtab`. Now you're ready to create and format the RAID device.

For example, take the RAID 5 configuration created in the previous section. The RAID device file is `/dev/md2`. You'll want to create the file and then format it. You can then mount the filesystem of your choice on that partition. If you want to have it mounted automatically the next time you boot Linux, you'll also need to incorporate it into `/etc/fstab`.

The following commands create and then format the RAID device:

```
# mkraid -R /dev/md2
# mkfs -j /dev/md2
```

WARNING *The* `mkraid -R` *`raiddevice`* *command deletes all data from all partitions associated with the* *`raiddevice`* *in* `/etc/raidtab`.

Mounting RAID

At this point, you're ready to mount your new RAID array on the filesystem of your choice. For example, if you want to set up RAID for your home directories, copy all files (including hidden files) from the `/home` directory to another location, mount your new RAID device on `/home`, and then restore the files. Assuming `/tmphome` exists, the following commands work for the `/dev/md2` RAID device just created:

```
# cp -r /home /tmphome
# mount /dev/md2 /home
# cp -r /tmphome/home /
```

Finally, to make the change permanent, label your new device and then add an appropriate entry in`/etc/fstab`. For the directory and device shown, you first run the `e2label /dev/md2 /home` command and then create a new entry such as the following in `/etc/fstab`:

```
LABEL=/home  /home   ext3  defaults  1 2
```

When you reboot, Linux should automatically mount the `/home` filesystem on your new RAID device, `/dev/md2`.

RAID Courtesy with *mdadm*

You don't have to reboot to implement a new RAID array; you can start it with the right `mdadm` command. For example, if you've just created a RAID 1 array with the `/dev/sda1`, `/dev/sdb1`, and `/dev/sdc1` partitions, you can create a `/dev/md0` array with the following command:

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
 ➥--spare-devices=1 /dev/sdc1
```

This command creates RAID device `/dev/md0`, at RAID 1 (`--level=1`), using two RAID formatted partitions (`--raid-devices=2`), with one spare partition. You can then mount the RAID `/dev/md0` device on the directory of your choice.

Summary

Computer data is fragile. Backups are important. Before you select a strategy for protecting your data, you need to consider various disaster scenarios. Standard scenarios range from the loss of a user's key file to complete data erasure and computer damage from an electromagnetic pulse.

Your response depends on the computers you need to protect. If you're a personal desktop user, most disasters are just an inconvenience. Chances are all you need to back up are files in your home directory. Backups of configuration files in `/etc` can save time as you reinstall and then reconfigure Linux. Administrators who are responsible for groups of computers need more complete backups. In some situations, you'll need media that you can access quickly, because information such as financial data can be time sensitive.

Three different types of backups are available: full, incremental, and differential. Full backups are complete backups of all files on entire computer systems. Differential backups include all data since the last full backup. Incremental backups include all data since the last backup of any type.

Wide varieties of media are suitable for backups. The main candidates are tape drives and writeable CDs/DVDs. If an individual tape or CD does not provide enough room, devices such as jukeboxes are available that collect large numbers of tape drives or CDs/DVDs in one backup computer.

For tapes and other media, you can use generic backup and restore commands such as `tar`, `cpio`, `dump`, and `restore`. If you're backing up to CDs or DVDs, you'll need to create an image of the files you want to save with the `mkisofs` command. Then you can write to the appropriate drive with the `cdrecord` or `dvdrecord` command.

One alternative to backups is RAID, which provides data redundancy. In other words, if any single hard drive fails, the right type of RAID ensures that no data is lost. Red Hat Enterprise Linux supports three types of RAID: RAID 0, which does not provide redundancy; RAID 1, which mirrors one hard disk onto another; and RAID 5, also known as striping with parity. Hardware RAID is available for this purpose.

In Red Hat Enterprise Linux, you can create a software RAID array from a series of partitions, formatted to the `Linux raid autodetect` file type. Once you've configured the device in `/etc/raidtab`, you can format and then mount your new RAID device. Just remember to label the partition with the `e2label` command. You also need to document that device in `/etc/fstab`, if you want Linux to mount it automatically the next time you boot.

In the next chapter, we'll start to examine Linux and networking in detail. Chapter 15 starts with a somewhat theoretical look at the TCP/IP protocol stack and IP addressing. It also gives you the tools that you need to configure private IP addresses on your LAN. This prepares you for future chapters, where you'll learn to manage Linux on your LAN, secure your Linux network, and more.

Part 4

Basic Linux Services

In this Part, you will learn:

- Chapter 15: A TCP/IP Primer
- Chapter 16: Managing Linux on Your LAN
- Chapter 17: Securing Your Linux Network

Chapter 15

A TCP/IP Primer

Many of the same people who developed the Unix operating system also worked on the network that would eventually become the Internet. They designed TCP/IP as the standard group of network protocols for this purpose. Because Linux is a clone of Unix, it is also customized for TCP/IP. However, TCP/IP is only one of several *protocol stacks* associated with modern networking.

TCP/IP is named for two of its component protocols, the Transport Communications Protocol and the Internet Protocol. TCP/IP actually includes several hundred individual protocols. Officially, it is known as the TCP/IP Protocol Suite.

Before we dig into the details of TCP/IP, we'll step back and take a look at the fundamentals of computer networks, both small and large. We need a way to identify every computer on a network, and a standard method of transferring data. Several other protocol stacks are available, and in this chapter we'll address two of them: NetBEUI and IPX/SPX.

NetBEUI is the NetBIOS Enhanced User Interface, developed by Microsoft and IBM. IPX/SPX is also named for two of its component protocols, Internetwork Packet Exchange and Sequenced Packet Exchange. Like TCP/IP, IPX/SPX includes a substantial number of individual protocols.

To help software designers develop different protocols, they needed specifications for standard levels of communication. Their agreements are documented through the International Organization for Standardization (ISO) as the OSI model of networking, where OSI stands for Open Standards Interconnection.

While the OSI model is often applied to TCP/IP, many designers subscribe to a conceptually similar four-level protocol stack. Many TCP/IP services would otherwise require software at several different OSI levels.

If you're not interested in all this theory, you can jump ahead to what you can do with TCP/IP, starting with IP addressing. Two versions of IP addresses are available. IP version 4 addressing is still in common use in the United States, but the newer IP version 6 addresses are coming into frequent use in other parts of the world. This chapter covers the following topics:

- Exploring network fundamentals
- Understanding protocol stacks
- Learning the basics of TCP/IP
- Using IP addressing

Exploring Network Fundamentals

A *network* consists of two or more computer systems set up to communicate with each other. To some extent, the "media" you use doesn't matter. You can set up a network using parallel cables, telephone modems, Ethernet cards, wireless adapters, or any other media that allow your computers to exchange information. If you can connect these computers directly or through a hub, you can set up a local area network (LAN). Each LAN typically has a special IP address known as a *network address*.

A LAN connects computers that are close to each other, such as within an office or a building. An internet consists of two or more connected LANs. Some internets are wide area networks (WAN). A WAN consists of two or more geographically separate networks. The biggest WAN is the Internet.

NOTE *Any network or group of networks that are managed by the same group is often known as a* domain. *For example, you could configure two separate networks,* `linux.sybex.com` *and* `windows.sybex.com`*; both would be part of the* `sybex.com` *domain.*

LANs and WANs

Linux LANs are usually configured to a standard known as IEEE 802.3, more popularly known as Ethernet. This type of network is much faster than a telephone modem. While standard Ethernet networks allow computers to communicate at speeds of 10 or 100Mbps (Ethernet and Fast Ethernet), Gigabit Ethernet (1000Mbps) is currently coming on line in many locations, and even faster networks (10Gbps Ethernet) are currently under development.

NOTE *Ethernet is actually a trade name. The proper name for this network is taken from the standard implemented by the Institute of Electrical and Electronics Engineers, IEEE 802.3. Fast Ethernet and Gigabit Ethernet are known by similar names, IEEE 802.3u and IEEE 802.3ae.*

But the distance between computers on an Ethernet is limited to a few hundred meters, depending on the type of connection. In essence, while the amount of area that a LAN can cover is limited, LANs are fast. In contrast, connections between LANs in a WAN can cover thousands of miles, but the speed of the connection is typically limited. Even "high-speed" WAN connections are typically limited to 1.4Mbps (the speed of a typical T1 line) or less.

NOTE *This speed limit on WANs is based on cost. Internet WAN "backbones" can carry tens of gigabits of data and are expensive to build; consequently, the associated "bandwidth" is shared among the customers of this WAN.*

The Internet

Even if you've never set up a network, chances are good that you already know something about networking from your experience with the Internet. When connecting to the Internet, most users and many Linux administrators work through an Internet Service Provider (ISP). If you're responsible for a larger network, you may have your own direct connection to the Internet and thus act as your own ISP.

You connect to the Internet through your ISP's gateway, which is a computer that connects that ISP to the rest of the Internet. When you search for a domain name, such as `www.mommabears.com`, your computer has to find the appropriate computer address. On the Internet, this is known as an *IP address,* which is usually stored on a Domain Name Service (DNS) server.

Domains

When you installed Red Hat Enterprise Linux, you may have entered a hostname, such as computer1, or a fully qualified domain name (FQDN), such as `linux1.mommabears.com`. Unless your computer serves information or otherwise directly connects to the Internet, the name you use does not matter. If you use a FQDN, make sure to use the same domain name when you install Linux on each of the computers on your network.

Alternatively, some ISPs may assign you a specific FQDN for your connection to the Internet. This is a common practice with higher speed connections such as cable modems or DSL (Digital Subscriber Line) adapters.

You can divide a domain into a number of subdomains. Each subdomain can represent a different LAN. For example, `linux.mommabears.com`, `windows.mommabears.com`, and `other.mommabears.com` can represent three different LANs.

Hostname

The alternative to an FQDN on a network is a hostname such as computer1. The FQDN of a computer includes the hostname and domain name, assembled together. For example, if your computer has a hostname of berkeley and your domain name is `california.now`, your fully qualified domain name is `berkeley.california.now`. Every hostname or FQDN is associated with a numeric address such as an IP address.

Hardware Address

Computers contact each other through the hardware address on their network cards. A hardware address may look like 00-60-08-8D-41-93. These are hexadecimal numbers, also known as *base 16*. Every network card built today is configured with a unique hexadecimal hardware address. When you configure a TCP/IP network, you associate an IP address with a hardware address.

NOTE *In hexadecimal notation, there are 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, and f.*

Understanding Protocol Stacks

Now you can see that computers on a network need different elements to communicate. They need domain names, numeric addresses, and hardware addresses. They also need connection managers and application protocols such as those related to mail, web pages, file servers, and more. These elements can be classified through a protocol stack.

A protocol stack is essentially a division of labor. Some protocols are associated with applications such as mail or DNS. Others cite domain names, IP addresses, and hardware addresses. Some can encrypt your data, manage the 1s and 0s of binary code, or govern a remote login session.

There are two major ways to divide this labor. One is known as the OSI model of networking. The following sections examine the basics of OSI, as well as a couple of the other major protocol stacks, NetBEUI and IPX/SPX. The other major model of networking is based on TCP/IP, and we discuss it later in this chapter.

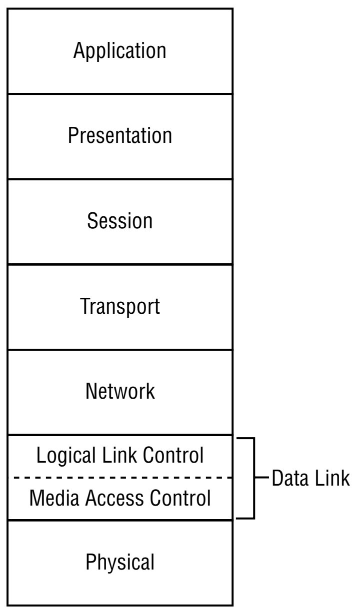
Tip *The arguments between supporters of the OSI and TCP/IP models of networking can be as vigorous as the arguments between the supporters of Linux and Microsoft Windows. While purists may object to the use of OSI to describe TCP/IP protocols, we believe that it is a useful learning exercise.*

OSI Levels

The OSI model of networking consists of seven levels. Before your computer sends a message over a network, your message is translated through these levels into the 1s and 0s that are actually sent over a network. The programs associated with each level perform different functions such as encryption, error checking, and routing.

The following is a brief description of each of these levels, from top to bottom, as shown in Figure 15.1. Pay attention to the numbers associated with each level.

Figure 15.1
The OSI model



Application Application-level (7) protocols start the translation process from the programs you use. For example, HTTP is an Application-level protocol that translates data from web browsers such as Mozilla. *Gateways* are computers that can translate applications between networks.

Presentation Presentation-level (6) protocols translate numbers and letters into lower-level computer code. One example is ASCII, which represents the numbers and characters on an English-language keyboard. Encryption protocols such as the Secure Sockets Layer (SSL) are also part of the Presentation level.

Session Session-level (5) protocols manage the time you spend on a network. These protocols determine which computer is sending and receiving messages at any particular time. For example, Session-level software in your network card determines whether data moves one direction at a time (half-duplex) or both directions simultaneously (full-duplex).

Transport Transport-level (4) protocols can resend your message until it gets a return receipt (a.k.a., an acknowledgment), or it can just send a message and make a best effort to get it to the destination computer. The two major TCP/IP Transport-level protocols are TCP and UDP. Transport protocols also begin breaking down messages into packets. TCP adds a request for acknowledgment to the start of the packet; UDP does not.

Network Network-level (3) protocols actually move the data from computer to computer and from network to network. IP is the quintessential Network-level protocol. Your messages need IP addresses to move between networks. Routers can manage traffic between networks at this level.

Data-Link Data-Link-level (2) protocols are primarily used to make sure your information gets to the destination computer correctly. This is often split into two sublevels: Logical Link Control (LLC) and Media Access Control (MAC). LLC protocols ensure that your messages reach the destination computer in order, without errors. This is also known as *frame synchronization* and *error checking*. MAC protocols help computers communicate with each other. That is why the hardware address of a network card is also known as a *MAC address*. Switches or bridges can manage traffic within a network at this level.

Physical Physical-level (1) protocols translate data into the 1s and 0s of computer communication. They also govern the physical world of networking, such as the cables and connectors.

NOTE *When you're shopping for network hardware, keep in mind that sales engineers often refer to components by a certain level. For example, standard switches work at level 2 and basic routers work at level 3. However, the boundaries are not rigid. For example, some switches include routing or transport functionality and are then advertised as "level 3" or "level 4" switches.*

The Life of a Packet

Unless your message is very small, computer networks don't send complete messages all at once. Starting at the Transport level, networks break messages down into packets. As you go further down OSI hierarchy, the packets may be further divided into smaller packets or even cells. Some protocols may send each packet or cell through a different route on the Internet; address information is included with each packet to make sure your message gets reassembled at the right computer, in order.

For example, Ethernet packets, which are created at the Data-Link level, can contain up to 1518 bytes. This includes 1500 bytes of data and 18 bytes of address information (and more), to ensure that the packet gets to the right computer on a network.

The details of network design are rich and complex. Perhaps the standard reference for network design is *Computer Networks*, by Andrew Tanenbaum (Prentice Hall, 2002).

NetBEUI

NetBEUI is the NetBIOS Extended User Interface. This is the set of protocols developed by Microsoft and IBM for networks. It is based on NetBIOS, the Network Basic Input Output System. NetBIOS includes a series of commands that allows a computer to send and receive data, as well as information on shared directories on other computers on that network.

The main drawback of NetBEUI and NetBIOS is that it is not routable. In other words, you can't connect a NetBEUI network to another network such as the Internet. A NetBEUI network is limited to 255 computers.

However, Microsoft has adapted NetBIOS commands to routable network protocol stacks such as TCP/IP and IPX/SPX. If you're an administrator of a network that includes Microsoft computers, you should know a few basic NetBIOS commands, such as `net view` and `net use`.

When you use Samba, you're taking advantage of the format associated with NetBIOS commands known as the Server Message Block (SMB). In Chapter 24, you'll learn about the Samba commands you can use on a Linux system. Since Samba is essentially the Linux/Unix implementation of NetBIOS, you should not be surprised to find Samba commands that correspond to NetBIOS commands such as `net view` and `net use`.

IPX/SPX

Like TCP/IP, IPX/SPX is actually a suite of protocols for network communication. It was developed by Novell, in support of its NetWare program, which is actually a network operating system.

Many older networks still use NetWare. However, NetWare also supports TCP/IP, so you probably don't need to adapt to IPX/SPX even if you're connecting to a NetWare-based network.

IPX/SPX is routable. In earlier versions of Microsoft Windows, IPX/SPX was the only choice available if you wanted to configure computers on multiple networks.

If you need to connect to an IPX/SPX network, you'll want the `mars-nwe-*`, `ipxutils-*` and `ncpfs-*` RPM packages. The first package allows your Linux computer to act as a file and print server on a NetWare network. The second package includes support for IPX/SPX. The final package includes the commands you need to act as a client on a NetWare network.

NOTE *There are several other major protocol suites, including IBM's System Network Architecture (SNA), the Xerox Network System (XNS), and the Digital Equipment Corporation network (DECnet). DEC is now part of Hewlett-Packard.*

Learning the Basics of TCP/IP

TCP/IP is the dominant network protocol suite today. Even Novell has been using it for years on its NetWare servers, and Microsoft uses TCP/IP even though it developed the rival NetBIOS suite. TCP/IP is the language of the Internet and is therefore generally the only protocol suite you need to know.

The TCP/IP Model

The TCP/IP model of networking includes four levels. The levels are roughly comparable to the OSI model. As shown in Figure 15.2, the TCP/IP Application level is somewhat functionally equivalent to the top three levels of the OSI model. The TCP/IP Link level is comparable to the bottom two levels of the OSI model.

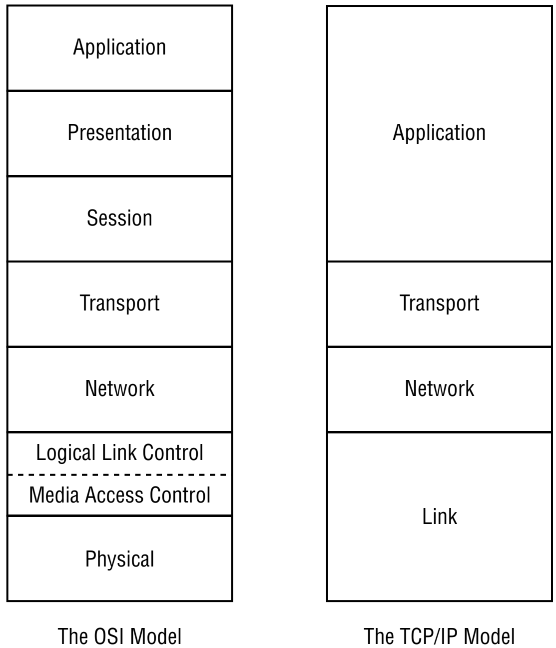


Figure 15.2
The TCP/IP model of networking

Naturally, the TCP/IP levels are better suited to different TCP/IP protocols. For example, Chapter 22 describes secure versions of FTP that manage communications between a client and server, which is an OSI Session-level function. They translate data into ASCII or binary code, which is an OSI Presentation-level function. And they translate your FTP commands, which is an OSI Application-level function.

Major Protocols

There are hundreds of TCP/IP protocols. You've probably heard of many of them, such as FTP, HTTP, SMTP, SNMP, TCP, IP, just to name a few. Some of these protocols are detailed in the following sections.

TCP/IP Application-Level Protocols

For a full list of TCP/IP Application-level protocols, see `/etc/services`. As shown in Figure 15.3, this file includes the name of a service, such as `ftp`, `ssh`, and `smtp`, the associated port number, and related comments.

TCP/IP has 65,536 available ports. Each port works conceptually like a TV channel. When you direct your Linux computer to the right port, you can receive the data associated with that port. The "well-known" ports are assigned by the Internet Assigned Numbers Authority (`www.iana.org`). Typical ports include 80 for HTTP (web pages), 21 for FTP communication, and 110 for POP3 e-mail.

FIGURE 15.3
/etc/services

```
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
smtp            25/tcp          mail
smtp            25/udp          mail
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/tcp          resource        # resource location
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
nameserver      42/udp          name            # IEN 116
nicname         43/tcp          whois
nicname         43/udp          whois
tacacs          49/tcp                          # Login Host Protocol (TACACS)
tacacs          49/udp                          # Login Host Protocol (TACACS)
re-mail-ck      50/tcp                          # Remote Mail Checking Protocol
re-mail-ck      50/udp                          # Remote Mail Checking Protocol
domain          53/tcp                          # name-domain server
```

Table 15.1 lists several important TCP/IP Application-level protocols and their associated ports.

TABLE 15.1: TCP/IP APPLICATION-LEVEL PROTOCOLS

PROTOCOL	PORT	DESCRIPTION
FTP	21	File Transfer Protocol; optimized for sending and receiving files
SSH	22	Secure Shell; encrypts communication between computers
Telnet	23	Connects in clear text to remote computers
SMTP	25	Simple mail transfer protocol for outgoing e-mail
HTTP	80	Hypertext Transfer Protocol for web pages
POP3	110	Post Office Protocol for receiving e-mail
SNMP	161	Simple Network Management Protocol for diagnosing networks
HTTPS	443	Secure HTTP
IPP	631	Internet Print Protocol, associated with the Common Unix Print System (CUPS)
SWAT	901	Samba web administration tool
NFS	2049	Network File Service for communication between Linux/Unix computers

TCP/IP TRANSPORT-LEVEL PROTOCOLS

By far, the two most important Transport-level protocols are TCP and UDP. Both take fully qualified domain names, such as `www.sybex.com`, and try to send your messages to those computers. TCP, also known as the Transmission Control Protocol, will keep sending a message until it gets an acknowledgment from the target computer. TCP is also known as a *connection-oriented* protocol.

On the other hand, UDP, also known as the User Datagram Protocol, does not need an acknowledgment. The assumption is that the network you're using is so reliable that any lost data doesn't really matter. UDP is also known as a *connectionless* protocol.

TCP/IP Network-Level Protocols

The key Network-layer protocol is IP, the Internet Protocol. This is most commonly associated with IP addresses such as 192.168.32.142. Both version 4 and version 6 IP addresses are discussed in detail toward the end of this chapter.

There is one other notable TCP/IP Network-layer protocol, the Internet Control Message Protocol (ICMP). This is most closely associated with the `ping` utility, which allows you to check the connection between your computer and every connected component on your network. You'll use `ping` and related utilities in Chapter 16.

NOTE *The TCP/IP Network level is also known as the* Internet level.

TCP/IP Link-Level Protocols

The TCP/IP Link-level protocols are most closely associated with networking technologies such as Ethernet, Token Ring, and ATM. This is where network packets are organized. Once organized, they are grouped into a stream of bits (1s and 0s). Next, the bits are sent through the network cable or other transmission media.

While the focus of networks today is on Ethernet, you may encounter several other important networking technologies. The following is just a short list of the available technologies:

Ethernet Regular Ethernet follows the IEEE 802.3 standard. It allows for data transfer at a theoretical maximum speed of 10Mbps. Because Ethernet packets wait to avoid collisions on a busy network, actual speeds are often less than half the maximum.

Fast Ethernet Fast Ethernet, which follows the IEEE 802.3u standard, allows for data transfer at a theoretical maximum speed of 100Mbps. It requires cables with a rating of Category 5 ("Cat 5") or better.

Gigabit Ethernet Gigabit Ethernet, which follows the IEEE 802.3ae standard, allows for data transfer at a theoretical maximum speed of 1000Mbps. It requires transmission media such as fiber-optic cables.

Token Ring Token Ring follows the IEEE 802.5 standard, which allows for data transfer at a theoretical maximum speed of 16Mbps. Since only the computer with the "token" is allowed to transmit data, it is more efficient than Ethernet, at least with respect to the maximum speed.

Asynchronous Transfer Mode (ATM) ATM networks are a popular option for higher-speed networks because they can transfer data at 155Mbps or 622Mbps. While support for ATM is considered to be "experimental," ATM network cards are explicitly listed in the Linux Hardware-HOWTO. Developers are working on creating ATM networks with transfer speeds faster than 2Gbps.

Point-to-Point Protocol (PPP) No discussion of networking protocols can be complete without reference to the protocol that has served us so well through regular telephone modems. While speeds are still limited to 56Kbps (53Kbps in the United States), PPP has served us well. And for those of you with high-speed Internet access, please remember that as of this writing, fewer than 20 percent of U.S. Internet users use "high-speed" services such as cable modems or DSL adapters.

NOTE *The TCP/IP Link level is also known as the* Network Access level.

Important Service Definitions

This section includes a basic list of major TCP/IP network services. If you are not too familiar with TCP/IP, this list can help you understand the services that are available. While you'll learn to configure some of these services in detail in later chapters, it can be useful to have a brief summary of each of the following services:

Domain Name System (DNS) The Domain Name System is a database of fully qualified domain names, such as `linux1.mommabears.com`, and IP addresses, such as 192.168.1.231. When you connect to the Internet and search for a site such as `www.redhat.com`, your Linux computer looks for a DNS server. Once it has an IP address, this information is added to your requests. Your message can then be sent from network to network until it reaches the Red Hat website.

Dynamic Host Configuration Protocol (DHCP) You can assign IP addresses to every computer on your network. But you need to be careful; if you accidentally assign the same IP address to two different computers, your network could fail. The Dynamic Host Configuration Protocol automates this process.

Address Resolution Protocol (ARP) The Address Resolution Protocol associates IP addresses with the hardware address of a computer's network card. These hardware addresses are also known as MAC addresses. Computers on a network communicate with hardware addresses. Your network can have problems if the IP address is assigned to the wrong MAC address.

Using IP Addressing

Every computer on a TCP/IP network needs an IP address before it can communicate with others. You or your ISP can assign a permanent address, or IP addresses can be "leased" from a DHCP server. Your ISP assigns your computer a unique IP address whenever you're connected to the Internet.

To set up IP addresses for your network, you need a network address and a network mask. IP addresses that share the same network address and network mask are on the same LAN. Network addresses fall into one of five address classes. Network masks define a range of IP addresses that you can assign with a specific network address.

Every network with a connection to other networks needs a gateway IP address for that connection. In Linux, you can limit access to and from your network with the `/etc/hosts.allow` and `/etc/hosts.deny` files or through appropriate `iptables` or `ipchains` firewall commands.

IP Version 4

The IP address standard in use since the 1970s is IP version 4 (IPv4), which is a 32-bit address. With 32 bits, there are more than 4 billion possible addresses ($2^{32} = 4{,}294{,}967{,}296$). That was more than enough addresses for the first years of the Internet. However, it isn't enough today. While the Internet is currently in transition to IP version 6 (IPv6), current IPv4 addresses will still be usable after the transition is complete.

In fact, IPv4 addresses are easier to understand and easier to configure for many private LANs. I think that IPv4 addresses will remain in common use for many years to come. In the next chapter, you'll learn how this allows you to configure private IP networks quickly and easily.

There are two ways to specify an IPv4 address: in binary notation, or in dotted decimal format. The following is a typical IPv4 private network address in binary notation:

```
11000000 10101000 00000001 00100000
```

Does this look confusing? Remember, this is the way computers read data. As humans, most of us are more familiar with the decimal system of numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. It's easy to convert bits into decimals: the previous IPv4 address in dotted-decimal format is 192.168.1.32. But not everyone can make this conversion so easily. It's worth taking a bit of time to understand how to convert bits of an IPv4 address to dotted decimal notation.

The Bits of an IPv4 Address

A bit is a binary digit. The binary system contains two possible numbers: 0 and 1. It's easy to represent a bit in a computer. All you need is a switch, an electrical impulse, or a pulse of light. When the switch is off, it's 0; when the switch is on, it's a 1.

By convention, there are 8 bits in a byte. In ASCII, every letter and number on an English language keyboard is associated with a unique byte. That's why a 32-bit IPv4 address is organized into four groups of 8 bits; this address has 4 bytes:

```
11000000 10101000 00000001 00100000
```

Now let's break down the bits in each byte. The first number in a byte, `00000001`, equals 1 in decimal notation. That's followed by `00000010` = 2, `00000011` = 3 and so on. Several examples of this are shown in Table 15.2.

Now let's take the first byte in the given address, `11000000`. That represents `10000000` = 128 and `01000000` = 64. Since 128 + 64 = 192, that's the first number in this IP address. The next number is `10101000`, which is 128 + 32 + 8 = 168. Similarly, `00000001` = 1 and `00010000` = 32, which leads to an IPv4 address of 192.168.1.32, expressed in dotted-decimal notation.

Taken to its logical extreme, note that `11111111` in binary notation = 255 in our numbers.

Table 15.2: Bytes and Regular Numbers

Byte	Regular number
00000000	0
00000001	1
00000010	2
00000100	4
00001000	8
00010000	16
00100000	32
01000000	64
10000000	128

Address Classes

IPv4 addresses range from 0.0.0.0 to 255.255.255.255. These addresses are divided into five address classes, A through E. You can assign IP addresses (when available) from Class A, B, or C. The range of addresses of each of the five different classes is shown in Table 15.3.

TABLE 15.3: IPV4 ADDRESS CLASSES

CLASS	RANGE	COMMENT
A	1.0.0.0 to 127.255.255.255	Allows networks of up to 16 million computers
B	128.0.0.0 to 191.255.255.255	Allows networks of up to 65,000 computers
C	192.0.0.0 to 223.255.255.255	Allows networks of up to 254 computers
D	224.0.0.0 to 239.255.255.255	Reserved for multicasts
E	240.0.0.0 to 255.255.255.255	Reserved for experiments

Not all of these IP addresses, even in classes A, B, and C, are usable. There are four types of addresses that you can't assign to a computer that is directly connected to the Internet:

- The first address in any network of IPv4 addresses is reserved as the network address.
- The last address in any network of IPv4 addresses is reserved as the broadcast address.
- The address 127.0.0.1 is reserved as the *loopback* address.
- There are groups of IPv4 addresses reserved as private addresses, suitable for private LANs that are connected to the Internet only through a firewall.

You'll learn about each of these addresses in detail in the next chapter, which will also cover the concepts of network and broadcast addresses, as well as network or subnet masks. These concepts will be covered in the context of a private IP network connected to the Internet.

IP Version 6

As strange as it sounds, 4 billion IPv4 addresses are not enough. All available IPv4 address groups have already been assigned. While you probably can get your own IPv4 address from your ISP (probably for an extra fee), work is under way to convert the Internet to IPv6.

An IPv6 address has 128 bits. That's more than 340,000,000,000,000,000,000,000,000,000,000,000,000 addresses. To ease the transition, a specific IPv6 address has been assigned to every IPv4 address. That leaves more than 3.4×10^{38} addresses for all other uses. Your IPv4 address will work in an IPv6 world.

The way IPv6 is configured, it's easy to convert an IPv4 address to IPv6. For example, the IPv4 address

```
192.168.1.32
```

is identical to the following IPv6 address:

```
::192.168.1.32
```

However, IPv6 addresses are also shown in hexadecimal notation. This is also known as base 16, where the numbers are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, and f. One example of an IPv6 address is as follows:

```
4aed:0a21:3c53:7dab:0000:0000:0000:0451
```

It's easy to convert IPv4 addresses to hexadecimal notation. As an example, convert the previous IPv4 address to binary format, like so:

```
11000000 10101000 00000001 00100000
```

Next, we know that 24 = 16. In other words, there are 4 bits in every hexadecimal number. Therefore, you should regroup the IPv4 address into groups of 4 bits (which is incidentally known as a *nibble*. . .no kidding).

```
1100 0000 1010 1000 0000 0001 0010 0000
```

Now, converting these numbers one at a time to decimal format leads to the following:

```
12 0 10 8 0 1 2 0
```

which equals the following in base 16 or hexadecimal format, like so:

```
c0a8:0120
```

The corresponding IPv6 address is as follows:

```
0000:0000:0000:0000:0000:0000:c0a8:0120
```

IP Version 6 Support

This section documents only the basic support provided by Red Hat Enterprise Linux 3 for IPv6. If you aren't familiar with Linux networking, some of the commands in this section may seem unfamiliar; we describe these commands in detail in several later chapters. For detailed information on how to use various IPv6 tools, please refer to *IPv6 Clearly Explained,*. by Pete Loshin.

Red Hat Enterprise Linux 3 supports IP version 4 (IPv4) and IP version 6 (IPv6) software by default. Note that IPv6 has come into common use in other parts of the world, especially in Europe. The Linux kernel already supports IPv6. If compiled properly (it is by default), you should be able to install the basic IPv6 module with the following command:

```
# modprobe ipv6
```

NOTE *Not all networking applications are supported by IPv6. The current status is maintained by the people behind the IPv6 HOWTO at* `www.deepspace6.net/docs/ipv6_status_page_apps.html`.

You'll need to use the IPv6 versions of various commands, such as `ping6`, `tracepath6`, and `traceroute6`. But in general, when available, IPv6 support is already built into the standard commands and daemons that we describe in this book.

You can find the basic IPv6 protocols listed in `/etc/services`; you'll want to add several IPv6 addresses to the following lines in `/etc/hosts`:

```
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
ff02::3    ip6-allhosts
```

You'll also need to make sure the `ipv6` module is loaded the next time you boot Linux; you can do so by adding the following line to `/etc/modules.conf`:

```
alias net-pf-10 ipv6
```

You can make your computer see IPv6 addresses now if you've run the `modprobe ipv6` command. As you can see in Figure 15.4, this includes a different IPv6 address.

FIGURE 15.4
A network card with IPv4 and IPv6 addresses

```
[root@Enterprise3 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:1C:BB:76
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1c:bb76/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:715 errors:0 dropped:0 overruns:0 frame:0
          TX packets:943 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:145349 (141.9 Kb)  TX bytes:838665 (819.0 Kb)
          Interrupt:10 Base address:0x10e0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:463 errors:0 dropped:0 overruns:0 frame:0
          TX packets:463 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43102 (42.0 Kb)  TX bytes:43102 (42.0 Kb)

[root@Enterprise3 root]#
```

Red Hat Enterprise Linux also supports firewalls using IPv6; the `iptables-ipv6` RPM is installed by default.

Summary

Unix was developed concurrently with the network that would eventually become the Internet. TCP/IP was developed as the language of the Internet. As a Unix clone, Linux is well suited to communicating on the Internet.

A network includes two or more computers set up to communicate with each other. While a LAN connects computers that are physically close to each other, WANs connect two or more geographically distant LANs. The largest WAN is the Internet. LANs are generally faster than WANs because of cost. In either case, you need to configure FQDNs, hostnames, IP addresses, and hardware addresses to communicate on any network.

Network languages such as TCP/IP are also known as *protocol stacks*. Major protocol stacks such as NetBEUI and IPX/SPX include dozens of protocols. Protocols are commonly classified in one of the seven levels associated with the OSI model of networking.

Because TCP/IP is the language of the Internet, it is the dominant network protocol suite. The TCP/IP model of networking includes four levels, which are better suited to describe the functionality of different TCP/IP protocols and services such as FTP, HTTP, SNMP, TCP, UDP, IP, Ethernet, and ATM. Other key TCP/IP network services include DNS, DHCP, and ARP.

Every computer that communicates on a TCP/IP network needs an IP address. The standard IP address system is IPv4. There are five IPv4 address classes. Since there aren't enough IPv4 addresses, we're currently in transition to IPv6. Nevertheless, IPv4 addresses are still in common use, especially since there is an IPv6 address available for every IPv4 address.

In the next chapter, you'll put these TCP/IP protocols and IP addresses to good use as you configure your computer and network. You'll also learn to connect your Linux LAN to the Internet.

Chapter 16

Managing Linux on Your LAN

Now that you've learned the networking theory in Chapter 15, you're ready to put that theory into practice on your Linux computer and network. For many of you, most of this chapter covers elementary concepts designed to help the future Linux administrators. If you've installed Red Hat Enterprise Linux using Kickstart (see Chapter 5), you may already be satisfied with your network configuration.

First you'll learn some of the basics of network hardware. Hubs connect the different computers in a LAN. Switches segment a LAN, which help you regulate traffic within your network. Routers serve as a junction between networks, directing traffic as needed.

Next, on a Linux computer, you need to configure your network card and make sure it's connected to the proper network card address by using the `ifconfig` and `arp` commands. Various commands are available to configure the hostname of your computer on a regular as well as a Network Information System (NIS)-based network. If you've set up Red Hat Enterprise Linux correctly, the appropriate network settings should show in files such as `/etc/hosts`, `/etc/host.conf`, `/etc/sysconfig/network`, and `/etc/resolv.conf`.

As we continue, you'll learn to configure a LAN with IPv4 private addresses. One reason why IPv4 addresses are still in common use is that they allow you to easily configure a LAN. With the right routing configuration and one public IPv4 address, you can connect this LAN to the Internet.

Red Hat Enterprise Linux includes some tools for connecting your computer to the Internet. While some are graphical, others require only the command-line interface. These tools include Red Hat's own Network Configuration Wizard and `minicom`.

Finally, if you have problems with your network, commands are available to help you troubleshoot any problems that may arise. The `netstat` command lets you measure traffic through different TCP/IP ports. The `ping` command enables you to check connectivity. And finally, the `traceroute` command helps you visualize the route that your messages may take through diverse networks, especially the Internet. This chapter covers the following topics:

- Understanding network hardware
- Configuring your computer on a LAN
- Configuring private and public networks
- Creating network connections
- Troubleshooting your network

Understanding Network Hardware

Before getting into how you configure Linux for a network, let's take a step back. Think about the physical layout of your network. While this is a book on Linux, most network problems are actually physical. Loose wires, unconnected cables, dust in hubs or routers, and similar issues are the most common causes of network problems. Based on the OSI model discussed in Chapter 15, you need to consider the following five categories of hardware on a LAN:

- Physical-level transmission media
- Physical-level hubs
- Data-Link-level switches
- Network-level routers
- Application-level gateways

Transmission Media

Your computer sends your data as 1s and 0s over *transmission media*. The data may be electrical impulses through copper wires, light pulses through fiber-optic cables, or even radio waves through the air. Transmission media work at the Physical layer of the OSI model.

Whatever means you use to transmit signals, there is a range limit. For example, an Ethernet network may not work as well as you hope if the length of twisted-pair copper cable between a computer and a hub is greater than the specified maximum cable length of 328 feet (100 meters). Briefly, here are some things to watch out for with physical media such as copper wires or fiber-optic cables:

Connections Check your connections. Many networks fail because cables are not properly plugged in.

Length Networks have a range. The standard "Category 5" network cable may not allow your Fast Ethernet network to perform up to capacity if your cables are longer than 100 meters (approximately 328 feet).

Installation Don't bend your cables too much. Severe bends can stretch parts of a cable, reducing their ability to carry data.

Hubs

A *hub* is the center of most modern LANs. Wired hubs are essentially boxes with sockets. With the right cable, you can connect a computer to each socket. When multiple computers are connected to a hub, the configuration looks like the spokes coming out of the center of a wheel, which is known as a *star* configuration. (I don't know why it isn't called a hub-and-spoke-configuration.)

Digital signals degrade with distance. A hub can rebuild a digital signal and retransmit it at its original strength. Because they just work with the 1s and 0s of computer communication, hubs also work at the Physical layer of the OSI model.

Switches

A *switch* is often used to split a larger LAN into two or more different logical network segments. Switches keep a database of hardware addresses on a LAN; in other words, they work at layer 2 of the OSI model.

Once first contact is made between two computers, they continue their conversation with their hardware addresses. Since switches know the hardware addresses on a LAN, they can retransmit every message (like a hub) and direct it toward the destination computer.

NOTE *Older switches are sometimes known as* bridges*. Both are designed at the Data-Link layer (2) of the OSI model.*

Routers

Routers transmit data between two or more LANs. A router has a network card on each of these LANs. In a TCP/IP network, each network card has an IP address. Thus, routers work at the Network layer of the OSI model.

In many cases, the gateway address that you configure in a file such as `/etc/sysconfig/network` should be the IP address of a router connected to your network.

Alternatively, you can configure a Linux computer as a router. First you need two or more network cards, connected to different networks. Then you must enable IP Forwarding in the kernel. It's easy to do with an IPv4 configuration by changing a setting in the `/proc` directory.

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

You can configure a router within a LAN, if needed, and it will perform the same functions as a switch or even a hub. To make sure this change is still there the next time you boot Linux, open the `/etc/sysctl.conf` file and verify that the following variable is set to 1:

```
net.ipv4.ip_forward = 1
```

Gateways

For most purposes, Linux assumes that routers and gateways are functionally equivalent. For example, if your network is connected to an outside network though an Ethernet network card `eth0` via a router, you can specify its connection to your LAN in the `ifcfg-eth0` file in the `/etc/sysconfig/networking/devices` directory as your `GATEWAY` IP address.

However, a *gateway* serves a different purpose, because it can connect LANs using different protocol stacks such as TCP/IP and IPX/SPX. It works at the OSI Application layer.

Configuring Your Computer on a LAN

While Red Hat Enterprise Linux usually configures your computer to connect to a LAN, you may want to change your configuration for various reasons. Say you have Linux on a laptop computer that you want to connect to another network. Or suppose you've acquired some computers from a different department. Or you're installing a second network card on your computer and need to make

sure the configuration of each network card is correct. Or perhaps Red Hat Enterprise Linux does not detect your network card.

Red Hat Enterprise Linux normally configures your network cards during the installation process. All you need is a detectable network card with a Linux driver and a Dynamic Host Configuration Protocol (DHCP) server. Alternatively, you can enter IP address and hostname information manually. If you're connecting to an NIS network, you may need to enter the appropriate names during the installation process. Chapter 23 covers NIS in more detail.

But when problems arise, it's important to know where to look to solve network configuration problems for your Linux computer. Some basic commands include `ifconfig` and `arp` (for configuring your network card) and various commands related to the hostname.

It's also useful to understand the basic network configuration files. The `/etc/sysconfig/network` file is just the start of a series of important Linux network configuration files.

Later in this chapter, we'll show you how to use minicom as well as Red Hat's Network Configuration Wizard to configure a number of different kinds of network connections.

Configuring with *ifconfig*

Perhaps the key Linux network configuration command is `ifconfig`, in the `/sbin` directory. With the right options, you can use this command to assign IP addresses, hardware ports, and network masks, as well as activate or deactivate a network card. It's easy to check your current network configuration. As shown in Figure 16.1, there are two active network components on my computer: an Ethernet card (`eth0`) and a loopback device (`lo`). As you can see, `eth0` includes connection information presumably for the LAN. The loopback device helps you make sure that Linux is properly connected to the TCP/IP protocol stack.

It's easy to assign a new IP address to your network card. The following command assigns the noted IP address to `eth1`:

```
# ifconfig eth1 10.122.238.3
```

FIGURE 16.1
ifconfig output

```
[root@Enterprise3 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:1C:BB:76
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1c:bb76/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2921 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4150 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:976136 (953.2 Kb)  TX bytes:473998 (462.8 Kb)
          Interrupt:10 Base address:0x10e0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:18846 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18846 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2004328 (1.9 Mb)  TX bytes:2004328 (1.9 Mb)

[root@Enterprise3 root]#
```

As discussed later in this chapter, the standard network mask for this IP address is 255.0.0.0. However, you can specify any network mask that you need with the new IP address:

```
# ifconfig eth1 netmask 255.255.255.0 10.122.238.3
```

Red Hat distributions have had problems in the past with assigning IRQ ports or I/O addresses to a *second* (or later) network card. While I no longer see the problem on my own computers, this may not be true for all hardware configurations. You can assign different hardware addresses to a network card. For example, the following commands assign IRQ 9 and I/O address 0x300 to the third Ethernet card on your computer:

```
# ifconfig eth1 irq 9
# ifconfig eth1 io_addr 0x300
```

As you can see in Figure 16.1, these settings correspond to the `Interrupt` and `Base address` settings in the output from `ifconfig`. If you see an error, the interrupt or address may already be assigned or reserved for plug and play.

You can use this command to activate or deactivate your network adapter. For example, the following commands deactivate and activate the `eth0` network adapter:

```
# ifconfig eth0 down
# ifconfig eth0 up
```

Configuring with *arp*

The Address Resolution Protocol (ARP) associates IP addresses with hardware addresses on a network card. Once your computer has made contact with another computer on your network, they exchange hardware addresses, which are then stored in an ARP database. Not surprisingly, you can find this database on your own computer by issuing the `arp` command (which identifies a problem):

```
# arp
Address      HWtype HWaddress          Flags Mask  Iface
192.168.7.2  ether  00:12:B5:64:3B:B2   C           eth0
Enterprise3  ether  00:60:0B:8A:41:93   C           eth0
192.168.7.2  ether  52:A5:CB:32:52:A2   C           eth0
allaccess    ether  00:20:78:09:D3:6A   C           eth0
```

Depending on how contact was made, the Address column lists either the IP address or the name of the remote computer. The computer name is taken from `/etc/hosts` for your convenience. The HWtype column shows the type of network adapter. The HWaddress column lists the hardware address of the adapter, in hexadecimal notation.

This particular output shows a duplicate IP address, which can stop communication on your network. You can remove the associated computer's entry in your ARP table by using the `arp -d` *`computername`* command. Be sure to substitute the name or IP address of the offending computer for *`computername`*.

The Hostname Commands

Several commands are available for defining or listing the name of your computer on various networks. These commands are illustrated in Table 16.1. With all but the `dnsdomainname` command, you can set the name of your computer. For example, the `hostname ilovehackers` command sets the name of your computer to `ilovehackers`.

Table 16.1: Hostname Commands

Command	Function
`hostname`	Lists or sets the hostname for the local computer
`domainname`	Lists or sets the NIS domain name
`dnsdomainname`	Lists the FQDN for the DNS server for your network
`nisdomainname`	See `domainname`
`ypdomainname`	See `domainname`

Network Configuration Files

Red Hat Enterprise Linux contains many important network configuration files. These include basic configuration files commonly used on other Linux distributions, such as `/etc/hosts`, `/etc/resolv.conf`, and `/etc/host.conf`. Red Hat Enterprise Linux also includes some newer configuration files that determine basic network settings in the `/etc/sysconfig` directory.

Tip *Red Hat is working toward consolidating configuration data, especially those related to network settings, in the* `/etc/sysconfig` *directory. If you're not sure where to look for configuration data, this directory is a good place to start.*

Static Hostnames—*/etc/hosts*

In the first days of the ARPAnet, only a handful of computers ran on this worldwide network. Those computers that were running Unix used the `/etc/hosts` file as a static database of computer names and IP addresses. Whenever a new university would join this network, it was relatively easy to change `/etc/hosts` and share a copy of this file with all computers.

While it is no longer practical to use `/etc/hosts` for the Internet, it is still a viable option for smaller networks. As long as you make sure that every computer on your LAN has the same copy of this file, it can serve your network well.

This file is fairly simple; each line includes an IP address, a fully qualified domain name (FQDN), and/or a hostname.

```
192.168.23.121   linux1.mommabears.com   linux1
```

DNS Servers—*/etc/resolv.conf*

The alternative to `/etc/hosts` is a Domain Name Service (DNS) server. In Linux, DNS is implemented through the Berkeley Internet Name Domain (`bind`), using the `named` daemon. (DNS is

covered in detail in Chapter 19.) If you have IP addresses for your DNS servers, you can enter them in the `/etc/resolv.conf` configuration file.

This is a simple file; every DNS server is known as a `nameserver`; this file associates it with an IP address. If you're connecting your network to an ISP, you can add the IP addresses of your ISP's DNS server to your file, in lines similar to this one:

```
nameserver 207.217.126.81
```

Search Order—*/etc/host.conf*

There are two databases of hostnames and IP addresses: `/etc/hosts` and DNS servers. The order is determined by `/etc/host.conf`. Normally, this file contains only one line:

```
order hosts,bind
```

This line configures your Linux computer to search for the right IP address in your `/etc/hosts` file, before checking `bind`, which, as described in the previous section, is the Linux name for a DNS server. You could even include an NIS server in this list; see the discussion on `/etc/nsswitch.conf` in Chapter 23 for more information.

Basic Network Settings—*/etc/sysconfig/network*

Basic network configuration data is listed in `/etc/sysconfig/network`. If you're having problems with your network, this is a good place to look. You should see the `NETWORKING=yes` line at the start of this file. Other variables are shown in Table 16.2. Not all of these variables are required in this configuration file; some are unneeded if you use a DHCP server.

Some of these variables may be located in network adapter–specific files in the `/etc/sysconfig/networking/devices` directory. For example, if you have more than one network adapter on your system, you may be connected to more than one network. In that case, the gateway address for each adapter may vary. Therefore, the `GATEWAY` variable would be associated with specific adapter configuration files such as `ifcfg-eth0`.

Table 16.2: */etc/sysconfig/network* Variables

Variable	Description
`NETWORKING`	This is `yes` or `no`; `yes` is required to let Red Hat run networking.
`HOSTNAME`	The hostname name of your computer.
`GATEWAY`	The gateway IP address of your computer.
`GATEWAYDEV`	The network device, such as `eth1`, that is connected to the network with the gateway; needed if you have more than one network card on the computer.
`NISDOMAIN`	The domain name of your NIS system, if available.

Configuring Private and Public Networks

In Chapter 15, you learned some of the basics of IPv4 addresses. Now you'll see how to make IPv4 addressing work in configuring a LAN that is connected to the Internet.

When you configure a network that's connected to the Internet, you can't select just any IP address. There are a number of *private* IP addresses that you can freely use on your internal network. However, for your connection to the Internet, you need at least one *public* IP address. Each of the computers on your network can access the Internet simultaneously using your public IP address.

Unfortunately, most public IP addresses are taken. Those that are still available are generally assigned by ISPs to their customers.

NOTE *Public IP addresses are used for communication between computers and networks on the Internet. On the other hand, the same private IP addresses can be used on independent private networks. To avoid confusion, private IP addresses are not valid for communication through the Internet.*

You can configure your LAN with private IP addresses, with one public IP address on a gateway computer for connecting your LAN to the Internet. To get a public IP address on the Internet, talk to your ISP. You'll get either a static IP address with a subnet or network mask or instructions to get your address from a DHCP server.

Network Definitions

Several basic terms define IP addresses on a LAN.

Network address Every IP address includes two parts: the network address and the numbers associated with a particular host. A network address such as 192.168.22.0 uniquely identifies a specific network. Assuming it is a Class C address, it identifies a network with a range of assignable IP addresses between 192.168.22.1 and 192.168.22.254.

Network mask This special IP address (also known as a *subnetwork mask* or a *subnet mask*) lets you define a range of available IP addresses on a LAN. The three "standard" network masks are 255.0.0.0, 255.255.0.0, and 255.255.255.0.

Broadcast address This is a special IP address used to communicate with all computers on that network. It is the last available IP address on a network. For example, if you have a network address of 192.168.22.0 and a network mask of 255.255.255.0, the broadcast address is 192.168.22.255.

Private IP address This is an IP address that is dedicated for private LANs. You can use a private IP address on a LAN that is connected to the Internet through a computer with a public IP address. The same private IP addresses are often used on different LANs. However, you aren't allowed to use a private IP address to connect directly to the Internet.

Public IP address This is an IP address that is used to communicate directly to the Internet.

Classless Inter-Domain Routing (CIDR) CIDR is a method of specifying nonstandard network masks. This allows you to subdivide or combine standard IP address ranges.

Private IP Networks

To set up the computers inside your network with private IP addresses, you need a network address and a network mask. These two parameters define a range of IP addresses. As described in Chapter 15, three standard ranges of private IP addresses are available, as shown in Table 16.3.

Table 16.3: Private IP Address Ranges

Range	Class	Description
10.0.0.1–10.255.255.254	A	Can accommodate about 16 million computers in one domain
172.168.0.1–172.168.255.254	B	Can accommodate about 65,000 computers in one domain
192.168.0.1–192.168.255.254	C	Can accommodate up to 254 computers in one domain

When you choose a network address and network mask, you typically choose a subset of one of the IP address groups shown in Table 16.3. For example, if you have a network address of 10.0.0.0 and a network mask of 255.255.255.0, the range of possible addresses is 10.0.0.0 through 10.0.0.255, which consists of 256 different addresses. These addresses compose a subnetwork, also known as a *subnet*.

But as you may remember from Chapter 15, the first address in this subnet, 10.0.0.0, is reserved as the network address. And the last address in this subnet, 10.0.0.255, is reserved as the broadcast address. You can't assign either address to a specific computer. That leaves 254 addresses on this subnet that you can assign to actual computers.

Note *Another private IP address range exists, the 169.254.0.0/255.255.0.0 network (with assignable addresses between 169.254.0.1 and 169.254.255.254). It's been assigned by the Internet Assigned Numbers Authority (*`www.iana.org`*) for computers without static IP addresses that can't get this information from a DHCP server.*

Network Mask

A network mask allows you to determine if a specific IP address is on the same LAN. It also enables you to differentiate network addresses from host addresses. When you put the network address together with the network mask, you can define the range of host addresses you can assign to your computers.

Table 16.4 shows several examples of network addresses, host addresses, and network masks. The Available Host Addresses column defines the IP addresses that you can assign on your internal network.

Table 16.4: Sample Network Addresses and Network Masks

Network Address	Network Mask	Available Host Addresses	Number of Assignable IP Addresses
10.0.0.0	255.0.0.0	10.0.0.1–10.255.255.254	16,777,214
10.21.92.0	255.255.255.0	10.21.92.1–10.21.92.254	254

Continued on next page

Table 16.4: Sample Network Addresses and Network Masks *(continued)*

Network Address	Network Mask	Available Host Addresses	Number of Assignable IP Addresses
10.182.0.0	255.255.0.0	10.182.0.1–10.182.255.254	65,534
172.168.78.0	255.255.255.0	172.168.78.1–172.168.78.254	254
172.168.0.0	255.255.0.0	172.168.0.1–172.168.255.254	65,534
192.168.3.0	255.255.255.0	192.168.3.1–192.168.3.254	254

From this information, you can derive the following “rules” for IP addressing:

- A network IP address is never used as a host address for a specific computer. This address comes just before the range of available host addresses.
- The 255s in a network mask normally correspond to the network address. For example, if your IP address is 10.162.4.23 and your network mask is 255.255.255.0, the network address is 10.162.4.0. The “host” part of the IP address is 23. See the section “Classless Inter-Domain Routing (CIDR)” for exceptions to this rule.
- The last address in an IP address range is reserved as the broadcast address. For example, for the last example in Table 16.4, the broadcast address is 192.168.3.255.
- Standard network masks are 255.0.0.0, 255.255.0.0, and 255.255.255.0. Other network masks are described in the section “Classless Inter-Domain Routing (CIDR).”

Configuring a Network

Before you set up TCP/IP on a LAN, you need to select a set of addresses. Based on the information in the previous sections, select a private network address and network mask. When you put the two addresses together, you get a range of IP addresses that you can assign to each computer on your LAN.

Perhaps the most common network mask is 255.255.255.0. As discussed earlier, this network mask allows you to choose from 254 IP addresses. In other words, if your network address is 10.168.0.0, this network mask allows you to assign 10.168.0.1, 10.168.0.2, 10.168.0.3, and so on, through 10.168.0.254 to different computers on your network.

Remember, the first address in the network range, in this case 10.168.0.0, is reserved as the network address. The last address in this range, 10.168.0.255, is reserved as the broadcast address.

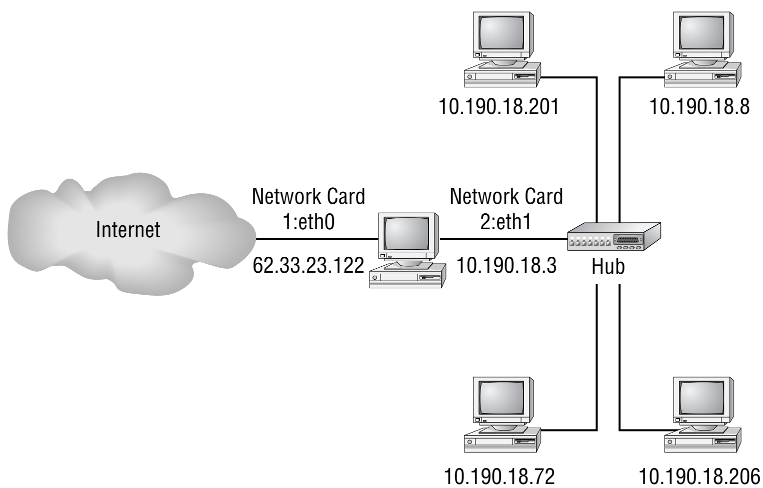
You have two choices with the assignable IP addresses. You can assign them to individual computers yourself, with commands such as `ifconfig` as described earlier. This means you also need to manually add the IP addresses for the DNS server and the default gateway. Alternatively, you can set up the range of available IP addresses on a DHCP server. As discussed in Chapter 19, DHCP servers can be configured to “lease” IP addresses to each computer on your network. That server can also pass along information related to the DNS server and gateway address for your network.

The Gateway Computer

On a network, the gateway computer is connected to your LAN and another network, such as the Internet. On a typical LAN, only one computer is directly connected to another network. That computer has two or more network cards: one is connected to the LAN, and the other is connected to the other network. One IP address is assigned to each network card. The gateway address is the IP address of the network card on the LAN.

To illustrate this configuration, look at Figure 16.2, which shows a LAN of five computers. The computer that is shown between the hub and the Internet is the gateway computer. The gateway address for all the other computers on this LAN is 10.190.18.3, which is the address that the gateway computer uses on the LAN.

Figure 16.2
Assigning IP addresses



The other network card on the gateway computer gets the public IP address on the Internet, in this case, 62.33.23.122.

Classless Inter-Domain Routing (CIDR)

Classless Inter-Domain Routing (CIDR) is not the easiest topic for speed-readers. However, if you take these explanations step by step, you'll be a CIDR master in no time at all.

In most cases, the only network masks you need on an IPv4 network are 255.0.0.0, 255.255.0.0, and 255.255.255.0. These network masks are most closely associated with Class A, B, and C addresses, respectively.

Those three network masks make it easy to differentiate a network address from the host address. For example, if one of the computers on a distant network has an IP address of 192.168.38.48, with a network mask of 255.255.255.0, you know the network address is 192.168.38.0. The computers on that LAN can have IP addresses between 192.168.38.1 and 192.168.38.254.

Bits and Bytes

To understand CIDR, you need to understand the bits and bytes in an IPv4 address. There are 32 bits in an IPv4 address. They are organized into 4 different numbers between 0 and 255, which correspond to 4 bytes. There are 8 bits in a byte. Each bit represents a different number. The top row represents the bits in a byte; the bottom row represents their decimal equivalent.

```
 1     1     1     1     1     1     1     1

128   64    32    16     8     4     2     1
```

For example, if you have a byte of 10000000, the corresponding number is 128. If you have a byte of 00010000, the corresponding number is 16. If your byte is 11111111, the corresponding number is 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255.

As an example, assume that you're setting up a Class C network, using the 192.168.38.0 network address. You may not even need 254 different IP addresses for your LAN; however, CIDR is useful if you're responsible for two LANs in separate buildings. In this case, you can use CIDR to subdivide IP addresses in a different way.

To understand how this works, let's take a step back and return to the bits. The following two IP addresses represent 192.168.38.48 and 255.255.255.0 in binary notation:

```
11000000 10101000 00100110 00110000
11111111 11111111 11111111 00000000
```

As discussed earlier, the 255s in a network mask correspond to the network IP address, in this case, 192.168.38.0. When expressed in bits, the 1s in a network mask correspond to the network address, as follows:

```
11000000 10101000 00100110 00000000
```

Note *Note how 255.255.255.0 corresponds to 24 bits of an IPv4 address. In CIDR notation, this network address and mask can be shown as 192.168.38.0/24.*

The last 8 bits are not "covered," which gives us a range of $2^8 = 256$ host addresses, starting with 0. The 0 is assigned as the end of the host network address; 255 is assigned as the host broadcast address. Neither of these addresses can be assigned to a specific computer; therefore, you have 254 addresses available on this LAN. Look at what happens when you add one more bit to the network mask:

```
11000000 10101000 00100110 00110000
11111111 11111111 11111111 10000000
```

The area "covered" by the 1s in the network mask represents the network address of 192.168.38.0. However, only the last 7 bits are not "covered," which gives you a theoretical range of $2^7 = 128$ host addresses, starting with 0 and ending with 127. Therefore, this particular network has an address of 192.168.38.0 and a broadcast address of 192.168.38.127. The network mask is 255.255.255.128.

NOTE *Observe how 255.255.255.128 corresponds to 25 bits of an IPv4 address. In CIDR notation, this network, with this network mask, can be represented by 192.168.38.0/25.*

Alternatively, look at the same network mask for an IP address of 192.168.38.166.

```
11000000 10101000 00100110 10110000
11111111 11111111 11111111 10000000
```

Using the same rationale, this particular network has an IP address of 192.168.38.128 and a broadcast address of 192.168.38.255. Remember, neither of these addresses can be used on a specific computer. Thus, there are only 126 available host addresses.

With a standard Class C network mask of 255.255.255.0, you can configure 254 computers on the 192.168.38.0 network. With a slightly different network mask (255.255.255.128), you can configure two different LANs with 126 available host addresses.

Creating Network Connections

We've already described how you can create a network connection using text commands such as `ifconfig`. In this section, we'll present several kinds of network connections, using the convenience of Red Hat's Network Configuration Tool. In many cases, you'll need this tool only when you add a network device after installation.

While we've focused this section on creating network connections between your LAN and the Internet, you can also use the techniques in this section to configure connections from individual computers inside your network.

In many cases, you'll have already configured networking when you installed Red Hat Enterprise Linux. Once configured, you can also use the techniques in this chapter to modify each computer's network settings as needed.

Even in the United States, the cost of higher-speed connections has come down to the point where it is cost-effective for most small businesses that need Internet access. High-speed Internet connections are also known as *broadband*. In most cases, Red Hat Enterprise Linux users will be connecting to the Internet using some sort of broadband connection. When you do so, you're essentially connecting your computer to your ISP's network.

Several broadband connection services are available, including satellite, infrared, wireless, cable modems, and DSL (Digital Subscriber Line) services. These services transmit and receive data at 144Kbps and higher speeds.

In most cases, connecting to a broadband service is no different from connecting your computer to a router. The ISP may provide, sell, or rent you a router. Either you connect to its DHCP server using the techniques described in Chapter 19 or you are given the IP address for your gateway and DNS servers.

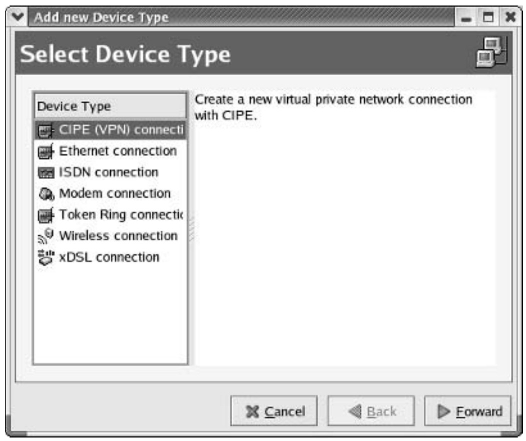
Of course, you can use Linux to connect to many ISPs with a regular telephone modem. Perhaps the best representative of a Linux text-based telephone modem interface is `minicom`. Red Hat has developed its Network Configuration Tool to guide you when you're creating a telephone modem or broadband connection.

The Red Hat Network Configuration Tool

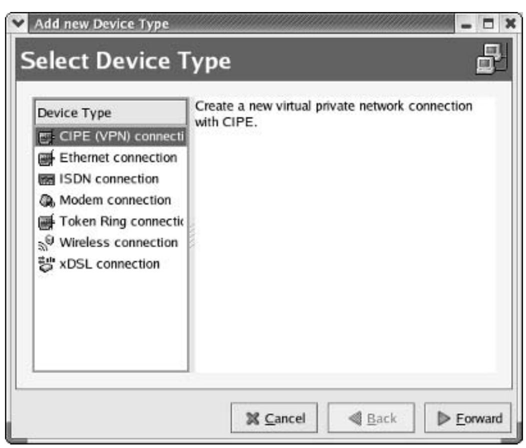
The two basic Red Hat graphical network configuration utilities are `redhat-config-network` and `redhat-config-network-druid`. These utilities control the configuration of network devices on your computer.

Start `redhat-config-network-druid`. Run that command from a graphical command-line interface, or select Main Menu ➢ System Tools ➢ Internet Configuration Wizard. This opens the Add New Device Type window, shown in Figure 16.3.

FIGURE 16.3
Adding a new network device



The Red Hat Internet Configuration Wizard is different from the Microsoft tool of a similar name. You can start it with a console command in GNOME or KDE by issuing the `redhat-config-network-druid` command. As you can see, this opens the Add New Device Type window, with options that let you configure a variety of different network devices. While the focus of this section is on regular telephone modems, let's take a brief look at the other options.



CIPE (VPN) Connection Crypto IP Encapsulation (CIPE) is more commonly known as Virtual Private Networking (VPN), which involves building a secure network connection through a public network such as the Internet. This option allows you to set an IP address for each end of the connection as well as an appropriate encryption key. Alternatively, you can also configure VPN with an IPsec connection, which we describe shortly.

Ethernet Connection This allows you to specify a driver, a device name such as `eth1`, and resources such as an IRQ port, an I/O address, and DMA channels appropriate for this network adapter. These settings use the `ifconfig` command to help Linux detect and communicate with this adapter. You can also set the network adapter to get IP addressing information from a DHCP server or configure these settings yourself. If the DHCP server is on a remote network, you'll typically need to specify the BOOTP protocol.

ISDN Connection As with Ethernet connections, this option allows you to specify a driver and resources for an ISDN adapter. Because ISDN is most popular in Europe, the settings are customized for several different nation-states on that continent.

Token Ring Connection This is a front end similar to the Ethernet configuration option.

Wireless Connection This is a front end similar to the Ethernet configuration option. Extra settings allow you to set the appropriate wireless channel and/or encryption key for your network.

xDSL Connection Several types of DSL connections are available, which vary in upload and download speeds. In any case, this utility enables you to configure the connection for an Ethernet adapter, with a username and password for the broadband ISP. This should also work for most cable modem connections.

Text-Mode Network Configuration

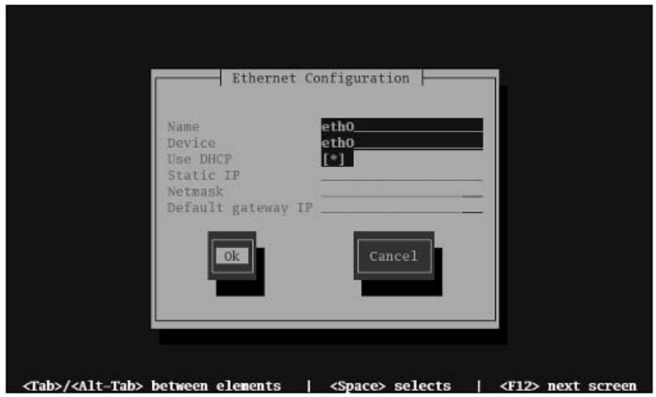
While not officially supported on Red Hat Enterprise Linux, you can configure networking on your Linux computer from a text-mode console. Start with the `redhat-config-network-tui` command. As you can see in Figure 16.4, this allows you to set up Ethernet, telephone modem, or ISDN adapter network devices.

FIGURE 16.4
Text-mode network configuration

NOTE *Red Hat does not officially support a number of "text-based" tools such as* `redhat-config-network-tui`*. If you need to administer a computer remotely, you can configure remote access to the X server. You can then use the Red Hat GUI tools to administer the server remotely. For more information on configuring remote X access, see Chapter 29.*

If you want to use this utility to set up an Ethernet connection, select that option. This takes you to the Ethernet Configuration window, shown in Figure 16.5. For more information on DHCP, see Chapter 19; if you want to set a static IP address, see Chapter 15 for more information.

FIGURE 16.5
Configuring an Ethernet card in text mode



As with the graphical tool, changes are saved in the `/etc/sysconfig/networking/devices` directory. The configuration file is `ifcfg-eth`*`n`*, where *`n`* is the number associated with the card.

If you want to use `redhat-config-network-tui` to set up a telephone modem, return to the screen shown in Figure 16.4. Select the Modem option. This takes you to the Modem Configuration window, shown in Figure 16.6.

FIGURE 16.6
Configuring a modem connection

You can set the name of your choice. If your modem was properly detected, it should be linked to `/dev/modem`. You should get the remaining information from your ISP; you normally do not need to complete the Modem Initstring field.

If your modem is working, changes are saved to the `/etc/sysconfig/networking/devices` directory. The configuration file is `ifcfg-ppp`*n*, where *n* is the number associated with the modem.

If you want to use `redhat-config-network-tui` to set up an ISDN adapter, return to the screen shown in Figure 16.4. Select the ISDN option. This takes you to the ISDN Configuration window, shown in Figure 16.7.

FIGURE 16.7
Configuring an ISDN adapter

Configure the adapter based on instructions from your ISP; this should include the Multiple Subscriber Number (MSN), normally provided by ISPs with ISDN services. If your ISDN adapter is working, changes are saved to the `/etc/sysconfig/networking/devices` directory. The configuration file is `ifcfg-isdn`*n*, where *n* is the number associated with the adapter.

NOTE *ISDN stands for the Integrated Services Digital Network, an older digital standard for telephones. Consumer ISDN adapters are more popular in Europe; they normally support data transmission rates of 128 or 144Kbps, depending on the system. (There are subtle variations between U.S., European, and Asian ISDN standards.)*

Setting Up a Network Adapter

In this section, we'll use the Red Hat GUI tool to configure a second (undetected) Ethernet connection. Return to the Red Hat GUI Network Configuration Tool.

NOTE *You can also configure network adapters through appropriate configuration files. Red Hat should automatically detect newly installed network hardware. For example, for a second Ethernet adapter, you can then configure network settings in the* `/etc/sysconfig` *directory, in the* `network`, `network-scripts/ifcfg-eth1`, `networking/profiles/default/ifcfg-eth1`, *and* `networking/devices/ifcfg-eth0` *files. However, with this many configuration files, the Red Hat tools can help you ensure that you make the appropriate changes to each of these files.*

Kudzu

If Red Hat Enterprise Linux did not detect your network card, try starting the Red Hat Hardware Discovery Utility, also known as kudzu. Sometimes kudzu can help you detect newly installed hardware, including network cards.

It runs automatically during the boot process. However, if you've just installed a new network card such as a PC Card in a laptop computer's PCMCIA slot, you may need to run kudzu again. If it finds something new, it will offer to configure the hardware for you, as shown here.

Run the `redhat-config-network` command if it isn't already open. Select Ethernet Connection, and click Forward to continue. This brings you to the window shown in Figure 16.8, where you can review the configured network card(s). If Linux has already detected all the network cards on your computer, you should be home free. Figure 16.8 assumes that Linux did not detect this card. Select Other Ethernet Card.

Now you can set up the device driver and hardware addresses associated with the new Ethernet card. Figure 16.9 allows you to specify the driver and hardware resources associated with the new card.

Finally, you can configure the network settings for the new card. As shown in Figure 16.10, you can use a DHCP or BOOTP server. (For more information on DHCP and BOOTP, see Chapter 24.) The Dialup option usually applies only if you're configuring a telephone modem and is associated with a telephone modem connection to an ISP.

FIGURE 16.8 Selecting an Ethernet card

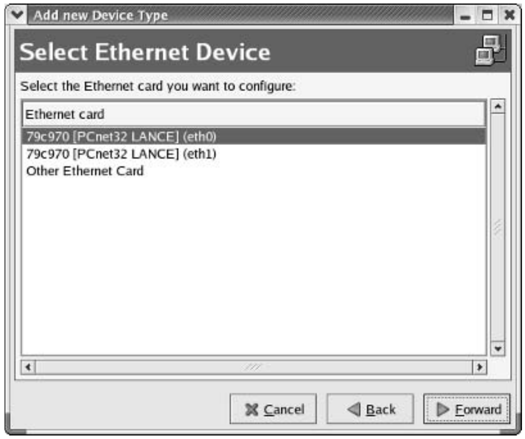


FIGURE 16.9 Specifying Ethernet adapter resources

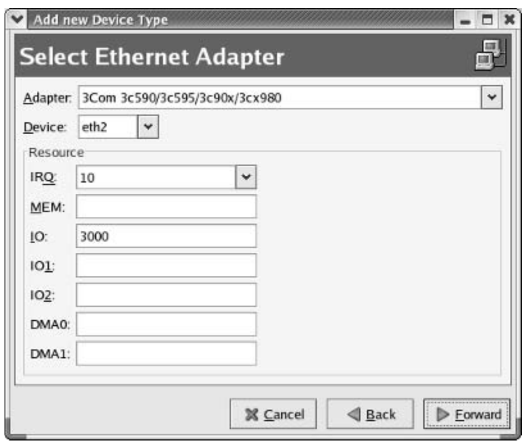
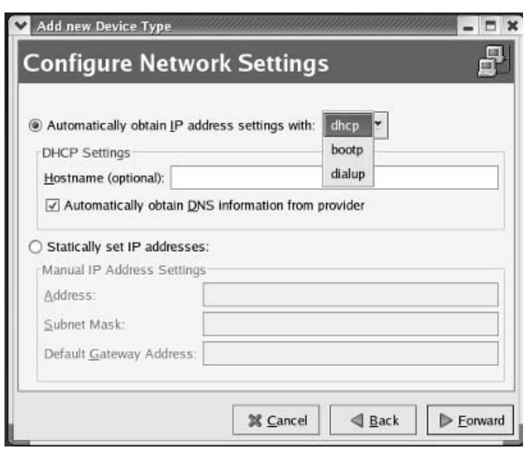


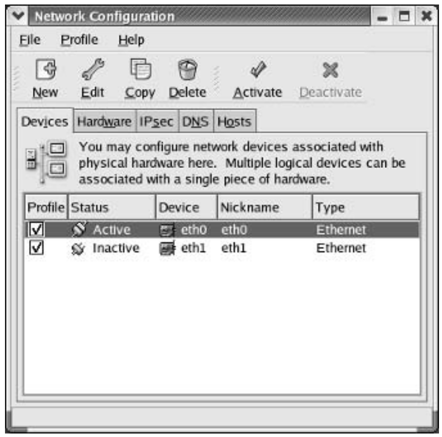
FIGURE 16.10 IP address settings



Alternatively, you can set up a static IP address. For more information on assigning IP addresses, see Chapter 15.

Once you've confirmed the changes, you're taken to the Network Configuration window, shown in Figure 16.11. This is the `redhat-config-network` tool, which you can also access by selecting Main Menu ➤ System Settings ➤ Network.

FIGURE 16.11
Managing a network configuration



When a network card is first configured, it is not active. You may activate it through the boot process, or you can highlight it and click the Activate button, as shown in Figure 16.11. If the configuration you set up is good, the displayed status will change from Inactive to Active.

The configuration for each network card is saved in the `/etc/sysconfig/networking/devices` directory. The configuration file for an Ethernet network card is `ifcfg-eth`*`n`*, where *`n`* is the number associated with the card.

SUPPLEMENTAL NETWORK CONFIGURATION

You can do more with the Network Configuration window shown in Figure 16.11. As you can see, the window contains four other tabs.

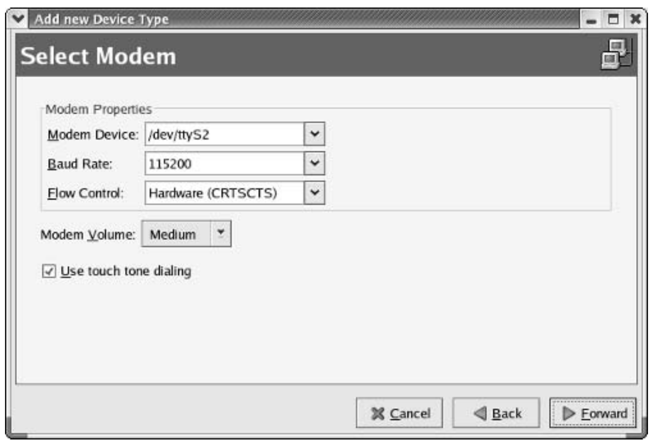
- The Hardware tab lists each configured network device.
- The IPSec tab allows you to create a secure connection between two computers; they can be local or separated by the Internet. These are sometimes also known as Virtual Private Network (VPN) connections. We'll describe this process (as well as CIPE connections) in more detail shortly.
- The DNS tab allows you to set the hostname for your computer, up to three different DNS servers, as well as a DNS search path. The hostname is saved in `/etc/sysconfig/network`; the DNS server information is saved in `/etc/resolv.conf`.
- The Hosts tab allows you to set up your own database of hostnames or domain names and their corresponding IP addresses. Changes you make are saved in `/etc/hosts`.

Configuring a Modem

Now let us move onto configuring a modem with Red Hat's Network Configuration Tool. One advantage is that it makes it easier for you to set up Red Hat's standard configuration files in the `/etc/sysconfig` directory. If you prefer to configure from the command-line interface, we describe how you can set up `minicom` shortly.

Incidentally, this configuration tool is also known as the Internet Configuration Wizard. Start it with the `redhat-network-config-druid` command. In the Device Type window, select Modem Connection. It checks your RPMs to make sure you have the necessary software. The wizard then tries to detect your modem. If it doesn't and you really do have a modem on your computer, refer to the discussion in Chapter 2 on Winmodems. Whatever the result, it will take you to the Select Modem window, shown in Figure 16.12, where you can configure the device, baud rate, sound, and other options for the modem. (It's normally a good idea to configure sound, so you can listen for a dial tone and characteristic modem sounds.)

Figure 16.12
Configuring a modem



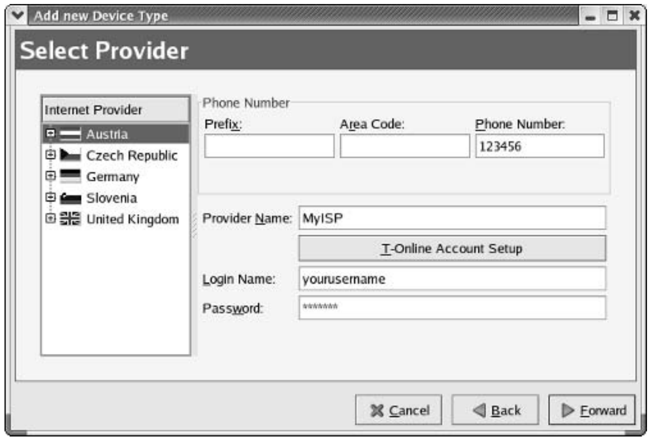
Note *Some Linux modem device files can be translated to Microsoft COM ports: for example,* `/dev/tty0`*=COM1,* `/dev/tty1`*=COM2, and so on. The modem detected in Figure 16.12 is detected on the device file associated with COM3. So if your modem worked as part of a Microsoft operating system, you may be able to find its COM port and use the corresponding Linux device. If your modem worked in Windows, but used COM5, a workaround can be found at linmodems.org.*

The baud rate should generally be two or four times the connection speed of your modem; for a 56Kbps modem (which is actually limited to a maximum of 53Kbps in the United States), you should normally select a baud rate of 115200 or 230400bps. Your modem will compress this data stream. In the Flow Control text box, you should generally leave the default, Hardware (CRTSCTS). When you're satisfied with the settings, click Forward.

Tip *To check the device associated with a detected modem, run the* `ls -l /dev/modem` *command. It should be linked to the actual modem device file,* `/dev/ttyx`*.*

In the next window, you can add the access number, login name, and password for your ISP, as shown in Figure 16.13. As long as you have this information, don't be concerned that your country is not on the Internet Provider list. (If you have a T-Online Account in Europe, click the T-Online Account Setup button and then fill in the prompts provided by T-Online, an ISP based in Germany.)

FIGURE 16.13
Adding ISP connection parameters



The next window is the IP Settings dialog box, shown in Figure 16.14. Normally, ISPs automatically provide IP address settings for a dial-up telephone modem connection. If your ISP has assigned you a static IP address, make sure you also have your assigned subnet mask and gateway address, and enter them here.

In the next window, click Apply. You should see the Network Configuration window, with settings for your network adapters. While the `ppp0` device shown in Figure 16.15 is "Inactive," all that means is that your modem isn't yet connected. Highlight your modem, and click Activate. If you enabled sound for your modem, you should hear it dialing your ISP.

FIGURE 16.14
Specifying IP settings

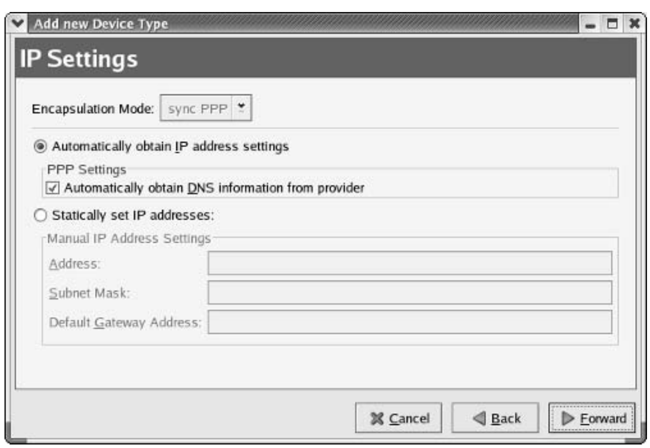
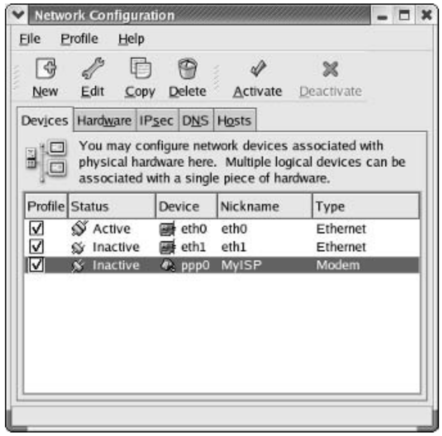


FIGURE 16.15
Activating your modem



When you're ready to drop your modem connection, return to the Network Configuration window, highlight your modem, and click Deactivate.

NOTE *If you activate your modem, and it still looks inactive in the Network Configuration Tool,run the* `ifconfig` *command. Your modem, normally device* `ppp0`*, may already be active.*

Using *minicom*

One traditional command-line tool for modem connections is `minicom`. You can start by configuring this utility as the root user with the `minicom -s` command. This starts the `minicom` Configuration menu, shown in Figure 16.16.

Before you can use `minicom`, you need to configure it to connect to your modem. Select the Serial Port Setup menu. You should see the menu shown in Figure 16.17.

FIGURE 16.16
Configuring minicom

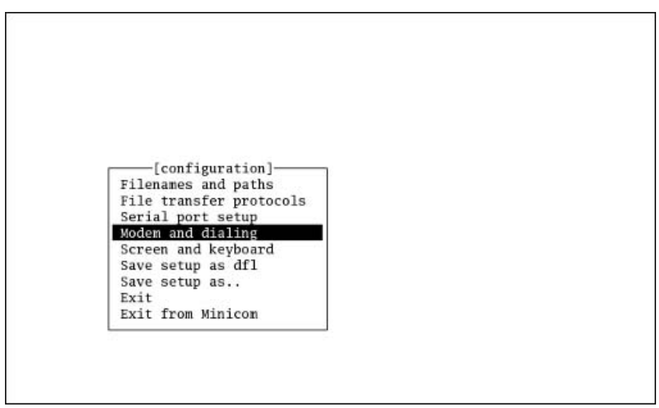
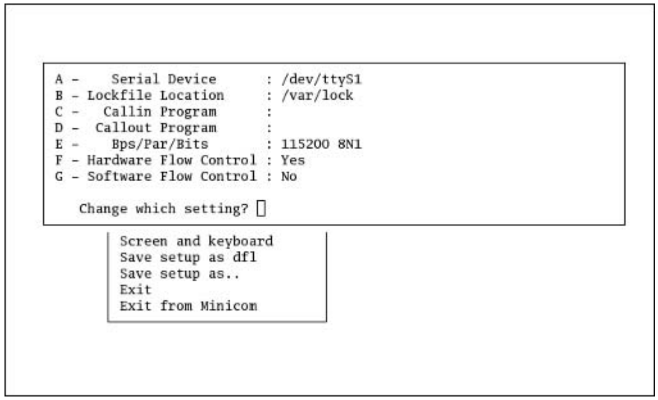


FIGURE 16.17
Configuring the serial port



```
A -    Serial Device      : /dev/ttyS1
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Bps/Par/Bits       : 115200 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No

   Change which setting?

        Screen and keyboard
        Save setup as dfl
        Save setup as..
        Exit
        Exit from Minicom
```

Depending on the modem, you may need to change the following settings:

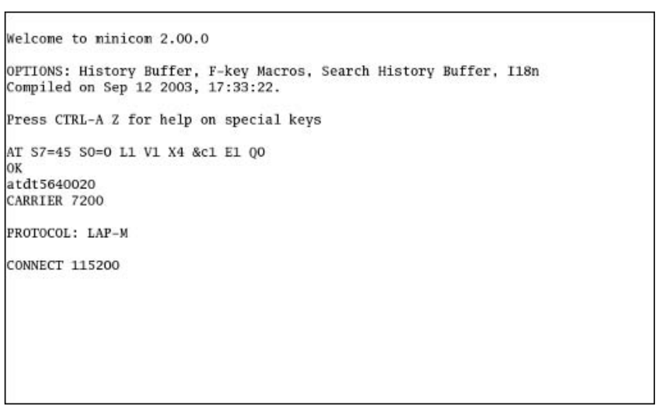
Serial Device The device associated with your modem. If an `ls -l /dev/modem` command reveals a link to a device such as `/dev/ttyS0`, use that device. Otherwise, some trial and error may be required.

Bps/Par/Bits Data settings for your modem. The bits per second (Bps) data rate should be two to four times the speed of your modem, because current modems compress data. Unless you have an older modem, the parity (Par) and stop bit (Bits) should match the default, 8N1.

Check your modem's documentation for any other settings, such as bps or hardware flow control, that you may need to change. Once configuration is complete, be sure to select Save Setup As dfl from the original menu. To start troubleshooting your modem, select Exit (not Exit from Minicom). This initializes your modem and brings you to a main `minicom` screen.

The most straightforward test is to try to dial your ISP. To do so, just enter the `atdt` command followed by the number of your ISP. For an example of this process, see Figure 16.18.

FIGURE 16.18
Connecting with minicom

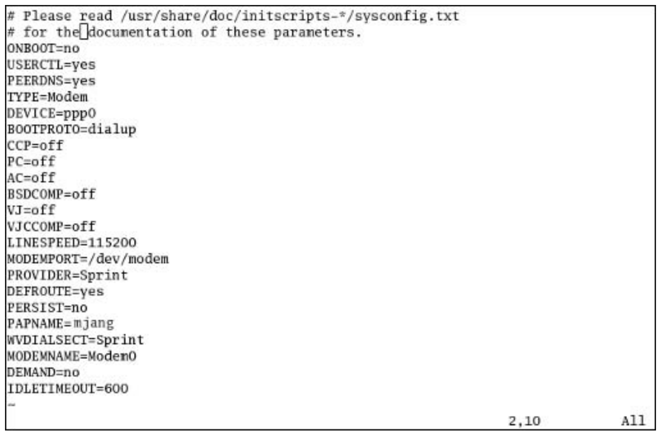


```
Welcome to minicom 2.00.0

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Sep 12 2003, 17:33:22.

Press CTRL-A Z for help on special keys

AT S7=45 S0=0 L1 V1 X4 &c1 E1 Q0
OK
atdt5640020
CARRIER 7200

PROTOCOL: LAP-M

CONNECT 115200
```

NOTE *The* `minicom` *utility uses common commands associated with terminal modem software. For example,* `atdt` *is short for "Attention, use Touch-Tone dialing."*

At this point, you may need to activate the `ppp0` modem device. You'll need to set up the appropriate configuration files in the `/etc/sysconfig` directory; it's easiest to follow the process we described previously to set this up. Otherwise, you'll have to configure parameters such as those shown in Figure 16.19. I've described some of the active variables from this `ifcfg-MyISP` file from the `/etc/sysconfig/networking/devices` directory in Table 16.5.

FIGURE 16.19
Modem connection parameters



```
# Please read /usr/share/doc/initscripts-*/sysconfig.txt
# for the documentation of these parameters.
ONBOOT=no
USERCTL=yes
PEERDNS=yes
TYPE=Modem
DEVICE=ppp0
BOOTPROTO=dialup
CCP=off
PC=off
AC=off
BSDCOMP=off
VJ=off
VJCCOMP=off
LINESPEED=115200
MODEMPORT=/dev/modem
PROVIDER=Sprint
DEFROUTE=yes
PERSIST=no
PAPNAME=mjang
WVDIALSECT=Sprint
MODEMNAME=Modem0
DEMAND=no
IDLETIMEOUT=600
~
                                                    2,10          All
```

TABLE 16.5: MODEM CONNECTION VARIABLES

VARIABLE	DESCRIPTION
`ONBOOT`	Whether to activate this device when Linux boots on your system; normally, this is `no` for a modem.
`USERCTL`	Determines whether regular users can activate this device.
`PEERDNS`	Specifies DNS information from a remote service, such as one run by an ISP.
`TYPE`	Notes the type of connection, such as modem or Ethernet.
`DEVICE`	Set to the name of the device for this network adapter, typically `ppp0` or `eth0`.
`BOOTPROTO`	Typically set to the DHCP server source; inactive if you have a static IP address.
`LINESPEED`	Specifies the transmission speed to the modem cable.
`MODEMPORT`	Set to the standard modem device, `/dev/modem`.
`PROVIDER`	Notes the ISP that you specified during the configuration process.
`PAPNAME`	Notes the login name, using the Password Authentication Protocol (PAP).

Continued on next page

Table 16.5: Modem Connection Variables *(continued)*

Variable	Description
`NETMASK`	Notes the network mask associated with this particular network.
`IPADDR`	Sets a static IP address for the local network.
`GATEWAY`	Specifies a gateway address for communication outside your network.

Once your connection is made, and the `ppp0` device is active (which you can verify with the `ifconfig` command), you can navigate normally on the connection, limited only by the speed of your telephone modem.

Virtual Private Network Connections

Some of the largest businesses have their own secure network connections between their geographically disparate locations. In that way, they have created their own Wide Area Network (WAN). However, dedicated lines over long distances can be quite expensive.

One alternative for businesses with fewer resources is to connect their disparate LANs over the Internet. There are three basic ways to configure this connection securely. One method is the Secure Shell (SSH), which we cover in Chapter 18. Two other methods supported by Red Hat Enterprise Linux are forms of Virtual Private Networking (VPN): Crypto IP Encapsulation (CIPE) and the IP security protocol (IPsec). I cover only CIPE in this book, because it can stay behind a network firewall.

NOTE *One drawback of IPsec is that it requires a dedicated server outside a firewall, which means that at least one computer on your network may not be secure. For more information on IPsec, see* `www.netbsd.org/Documentation/network/ipsec/`.

Crypto IP Encapsulation (CIPE)

CIPE was developed for Linux. You can configure it as if it were another network device, to support connections between two private LANs through the Internet. It encapsulates data in UDP packets. To make it work, you'll need to install the `cipe` RPM and configure the firewall on each network to accept UDP packets, as well as information from the CIPE device.

NOTE *If you're connecting two geographically distant LANs, chances are good that both are configured on private IP networks as described in Chapter 15. Make sure that each LAN on your networks has a different private IP network address. Otherwise, it may not be possible to create a VPN connection between your networks.*

There are a number of sample configuration files available in the `/usr/share/doc/cipe-1.4.5-16/samples` directory. Once configured, you'll want to copy the content of *some* of these files to the `/etc/cipe` directory. There are six files in this directory, as described in Table 16.6. Remember, you need two to tango in any connection; you'll need to configure these files on a CIPE client and a server.

NOTE *When I cite specific version numbers such as* `cipe-1.4.5-16`*, I'm citing the information I have at the time of this writing. The version you see depends on the updates you have installed.*

Table 16.6: CIPE Configuration Files

File	Description
`ip-up`	Activates the CIPE device; the default already in `/etc/cipe` is normally good enough.
`ip-down`	Deactivates the CIPE device; the default already in `/etc/cipe` is normally good enough.
`options`	Basic CIPE configuration options; includes private IP address and public domain names. Copy to `/etc/cipe` with the device name; the first CIPE device would be `options.cipbc0`.
`redhat-ifcfg-cipcb0`	Basic network configuration options; includes port numbers and local and remote connection addresses. Copy to `/etc/sysconfig/network-scripts` with the device name; the first CIPE device would be `cipcb0`, and the associated configuration file is `ifcfg-cipcb0`.
`redhat-options. cipcb0`	Sample encryption key; no need to copy this file, as the encryption key is already documented with other parameters in `/etc/cipe/options.cipbc0`.

CIPE Encryption Key

For everything to work, you'll need to make sure that every client and server with a CIPE connection includes the same encryption key. You can use the Linux random number generator device (`/dev/random`) to create an appropriate 128-bit key in hexadecimal notation with the following command:

```
# od -N 16 /dev/random -t x4 | awk '{print $2 $3 $4 $5}
```

CIPE Options on the Client and Server

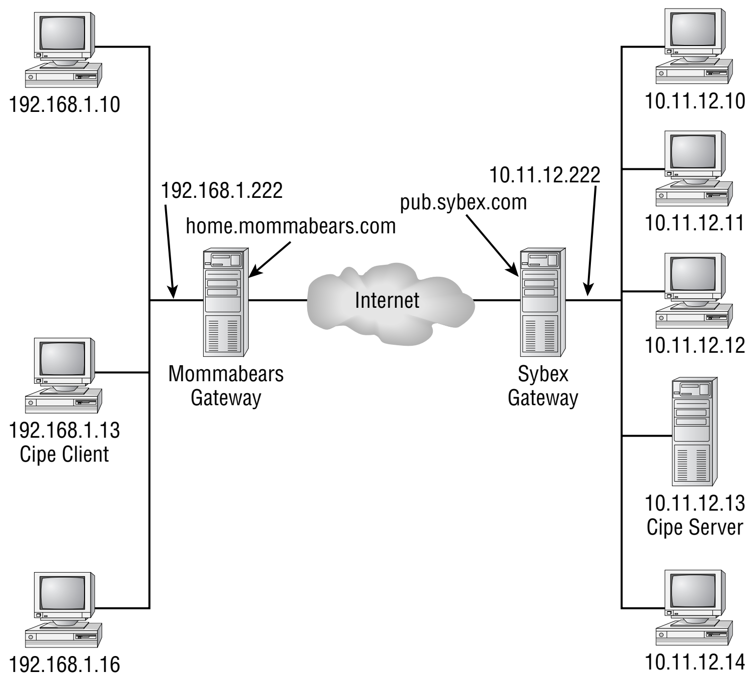
You can have multiple CIPE connection files, starting with `cipbc0`. For the first CIPE connection, start with the sample `options` file. Once you're finished, copy that file to `/etc/cipe/options.cipbc0`. This file is straightforward—it defines the client and server, using their IP addresses and their LANs associated public domain names. Finally, it includes the encryption key. This corresponds to five variables, as shown in Table 16.7.

Table 16.7: CIPE Options' File Variables

Variable	Description
`ptpaddr`	The IP address of the CIPE server device
`ipaddr`	The IP address of the CIPE client device
`me`	The domain name and port of the local gateway
`peer`	The domain name and port of the remote LAN gateway
`key`	The CIPE encryption key

As an example, take a look at Figure 16.20, where you're trying to connect from the 192.168.1.0 LAN to the 10.11.12.0 LAN.

FIGURE 16.20 A sample LAN to LAN CIPE connection



For the case shown, you would include the following information in your client `/etc/cipe/options.cipcb0` file:

```
ptpaddr 10.11.12.13
ipaddr 192.168.1.13
me home.mommabears.com:6789
peer pub.sybex.com:6543
key 3248fd20adf9c00ccf9ecc2393bb3e4
```

On the server, naturally this file will be slightly different:

```
ptpaddr 192.168.1.13
ipaddr 10.11.12.13
me pub.sybex.com:6543
peer home.mommabears.com:6789
key 3248fd20adf9c00ccf9ecc2393bb3e4
```

Notice how the address information on the server version of this file is opposite to the client; but the identical encryption key is used on both ends of the connection. Naturally, the firewall on each LAN needs to allow CIPE messages through the firewall; generally that requires that you set your network to accept UDP packets from the remote network. Based on Figure 16.20, the key `iptables` command that I'd add to my mommabears.com firewall would be as follows:

```
iptables -A INPUT -j ACCEPT -p udp -s 10.11.12.13
```

It's acceptable to use a private IP address in this case, as long as there is no conflict between the private IP address and the IP network addresses that you've used to connect through the Internet. For more information on firewalls and `iptables`, see Chapter 17.

Naturally, you'll need to specify a route. You can use the `route` command, described shortly. For now, the basic command that I'd add to the CIPE client computer on the mommabears.com network (see Figure 16.21) is as follows:

```
# /sbin/route add -net 10.11.12.0 netmask 255.255.255.0 gw 192.168.1.222
```

To make sure this command runs the next time you boot Linux, add this command to the `/etc/rc.local` file. You'll need the complete path (`/sbin/route`) in that file.

If you use Red Hat tools to configure a CIPE connection, you'll see different variables in different configuration files. For example, the `/etc/cipe/options.cipcb0` file created by the Red Hat tool includes the encryption key, a `cttl` (Carrier Time To Live), and a `maxerr` variable. The other variables are incorporated into other configuration files.

CIPE Start Script

With this information, you'll want to configure a start script in the `/etc/sysconfig/network-scripts` directory. As described in Table 16.6, the first CIPE connection script would be `ifcfg-cipcb0`. This file is straightforward; it normally includes the following commands:

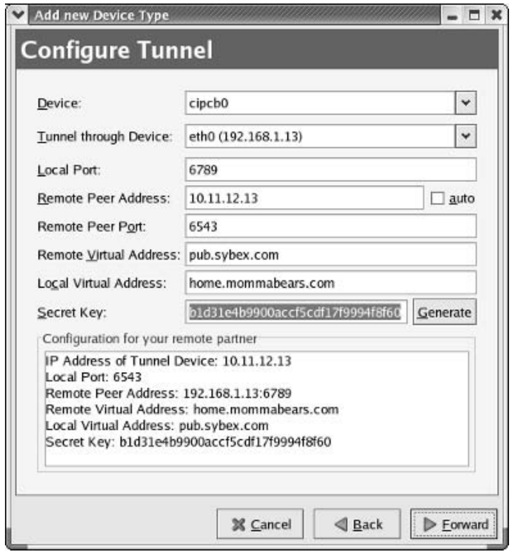
```
DEVICE=cipcb0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

As you can see, this specifies the CIPE device, it starts the connection when you boot Linux, and it does not require a separate IP address (`BOOTPROTO=none`), with control limited to the root user (`USERCTL=no`).

CIPE GUI Configuration

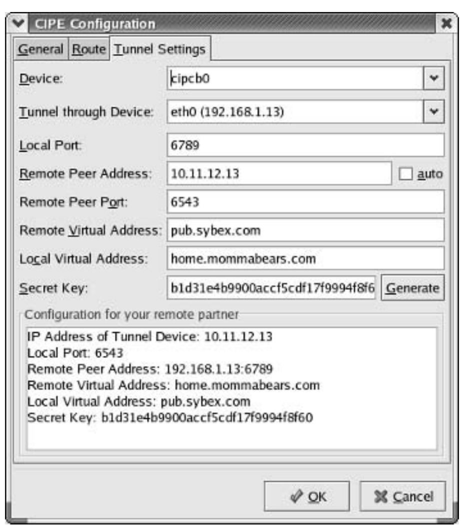
If this is all too much for you to handle, you can configure a CIPE connection using the Red Hat Network Configuration Tool. You can run the `redhat-config-network-druid` command to return to the selections shown in Figure 16.3. Specify CIPE (VPN) Connection. You can specify the tunnel that you'll configure, as shown in Figure 16.21.

FIGURE 16.21
CIPE configuration using the Red Hat tool



Once configuration is complete, you'll see the CIPE device in the Network Configuration window. You can then add the route; highlight the device, and click Edit. You'll see three tabs, as shown in Figure 16.22. The General tab allows you to make this device start the next time you boot Linux. The Route tab lets you configure the route between your client CIPE computer and the CIPE server on the remote network. And the Tunnel Settings tab allows you to change the configuration as required.

FIGURE 16.22
CIPE details using the Red Hat tool



NOTE *You may notice that the configuration files created earlier and by the Red Hat tool are somewhat different. The Red Hat tool includes a number of defaults that aren't absolutely required in configuration files.*

Troubleshooting Your Network

We've discussed troubleshooting techniques throughout this book. Troubleshooting a network is no different. If you have a problem, collect data, identify and isolate the cause, research the symptoms with others, and if none of this works, apply the scientific method.

As noted earlier, the number one cause of network problems is physical: bad connections, cables, power, and so on. Once you've checked the physical problems, Linux has a number of troubleshooting commands that can help. While the `netstat` command allows you to collect data, the `ping` and `traceroute` commands help you isolate the problem.

Checking Network Status

There are two things you should do to check the status of your network. First, run the `ifconfig` command to make sure your network card is still active. As discussed earlier, you can run the `ifconfig eth0 up` command to activate the `eth0` network card. If your network card is working, the next step is to check the status of your network with the `netstat` command.

This command displays routing tables, proxy connections to outside networks, interface statistics, and more. For example, the `netstat -a` command displays all available connections. As shown in Figure 16.23, the Local Address column displays names and numbers, which correspond to TCP/IP ports described in earlier chapters. In the Foreign Address column, you can see Samba (`netbios-ssn`), `http`, and `ssh` connections that are established between the local computer and two others.

FIGURE 16.23
netstat -a output

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:32768                 *:*                     LISTEN
tcp        0      0 *:nfs                   *:*                     LISTEN
tcp        0      0 Enterprise3d:32769      *:*                     LISTEN
tcp        0      0 *:rsync                 *:*                     LISTEN
tcp        0      0 *:618                   *:*                     LISTEN
tcp        0      0 *:35147                 *:*                     LISTEN
tcp        0      0 *:netbios-ssn           *:*                     LISTEN
tcp        0      0 *:sunrpc                *:*                     LISTEN
tcp        0      0 *:http                  *:*                     LISTEN
tcp        0      0 *:x11                   *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 Enterprise3d:ipp        *:*                     LISTEN
tcp        0      0 Enterprise3d:smtp       *:*                     LISTEN
tcp        0      0 *:https                 *:*                     LISTEN
tcp        0      0 *:637                   *:*                     LISTEN
tcp        0      0 *:microsoft-ds          *:*                     LISTEN
tcp        0      0 Enterprise3d:ipp        Enterprise3d:39969      ESTABLISHED
tcp        0      0 Enterprise3d:39969      Enterprise3d:ipp        ESTABLISHED
tcp        0      0 192.168.1.4:netbios-ssn bluesman:3485           ESTABLISHED
tcp        0      0 192.168.1.4:netbios-ssn allaccess:3107          ESTABLISHED
tcp        0      0 192.168.1.4:39997       bluesman:microsoft-ds   ESTABLISHED
tcp        0      0 192.168.1.4:netbios-ssn bluesman:1027           ESTABLISHED
tcp        0      0 192.168.1.4:ssh         bluesman:4015           ESTABLISHED
:
```

A routing table lists currently configured paths from your computer to another computer on or outside your network. Linux uses these paths to find the computers to which you want to connect. A variation of `netstat` enables you to inspect your routing tables. We've shown a fairly simple routing table in Figure 16.24. It includes three different types of IP addresses, as described in Table 16.8.

FIGURE 16.24
A routing table

```
[root@Enterprise3d root]# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         192.168.1.113   0.0.0.0         UG        0 0          0 eth0
[root@Enterprise3d root]# 
```

TABLE 16.8: A ROUTING TABLE

DESTINATION	COMMENT
192.168.1.0	No gateway is required for addresses on this network, since it is on the current LAN.
169.254.0.0	A default private network; used by computers without a static IP address *and* cannot get an IP address from a DHCP server.
127.0.0.0	No gateway is required for the loopback address, since it is on the local computer.
0.0.0.0	Use 192.168.0.113 as the default gateway for all IP addresses not specified earlier in this routing table. (You may see `default` in place of `0.0.0.0` in the Destination column.)

If needed, you can use the `route` command to add to your routing table. For example, assume you just added another LAN, with a network address of 10.0.0.0 and a network mask of 255.255.0.0, and connected it to a different network adapter, `eth1`. Add it to your routing table with the following command:

```
# route add -net 10.0.0.0 netmask 255.255.0.0 dev eth1
```

Checking Connections with *ping* and *traceroute*

If you have a specific problem on your network, such as a user who no longer has web or e-mail access, start by talking to the user. Based on your knowledge of browsers and e-mail managers, make sure the user knows how to access the desired service. You may also be able to log onto that user's computer through `ssh` or `telnet` and check the user's computer for yourself.

Linux includes a number of tools that allow you to work from the most basic network connection all the way to the connections required for the application. These command tools are based on the `ping` and `traceroute` commands. When diagnosing network connections, try the following commands. If they work, you'll need to press Ctrl+C to stop the response.

1. `ping 127.0.0.1`: This checks connectivity to the loopback address. If you see a continuous response such as `64 bytes from 127.0.0.1`..., TCP/IP is properly installed on your computer.

TIP One alternative to a continuous ping is the `ping -c 4 ip_address` *command, which sends four ping packets to the destination computer and then stops automatically. You can even set* `alias ping='ping -c 4'`*; for more information on the* `alias` *command, see Chapter 8.*

2. `ping your_ip_address`: Substitute the IP address defined for your network card for `your_ip_address`, based on the output from the `ifconfig` command. If you see a similar continuous response, your network card is properly configured.
3. `ping your_host_name`: Substitute the hostname for your computer, which usually can be found in `/etc/sysconfig/network`. If you see the same response as with the previous command, hostnames are properly configured on your computer.
4. `ping another_ip_address`: Substitute the IP address of another computer on your LAN for `another_ip_address`. You can use `ifconfig` to find IP addresses on Linux computers. (The corresponding Microsoft Windows command is `IPCONFIG`.) If you see a similar continuous response, communication is working on your LAN. You've at least configured those two computers with at least the same network address and netmask. As a follow-up, try the IP address for the default gateway on your LAN.
5. `ping another_hostname`: Substitute the name of a computer on a connected network for `another_hostname`. If you're connected to the Internet, one example is `ping www.Sybex.com`. If this works, your LAN's gateway or router is properly configured, and communication is possible to and from your LAN.
6. `traceroute another_hostname`: Use another name on a connected network. If you're connected to the Internet, run the `traceroute www.Sybex.com` command. Watch as you see the path your messages take from your computer to the Sybex website. If you're diagnosing a problem on interconnected networks, this command stops either at the destination or at the router or gateway that is having a problem.

For example, if you're having trouble with the `ping` command for the IP address of the router or gateway on your LAN, check the IP address of some other computer on your network. If you cannot connect to other computers on your LAN, there may be a problem with the cables or connections. Otherwise, it may be a problem with the hardware on the router computer.

Summary

In this chapter, you learned some of the basic steps required to configure a LAN with Linux. There are basic hardware components that you can use on a LAN. Transmission media usually involves copper wires or fiber-optic cables. Hubs connect computers on a LAN. Switches are often used to separate a LAN into segments. Routers can transmit data between two or more LANs. Gateways can even translate between different protocol stacks, such as TCP/IP and IPX/SPX.

For several reasons, you may need to change the network configuration that you set up during the Linux installation. Perhaps the key command to configure network cards is `ifconfig`. You can use `ifconfig` to assign hardware ports and IP address information. You can even use it to activate

or deactivate a network adapter. The `arp` command lets you check for duplicate IP addresses. The hostname commands allow you to set the name of your computer as seen by various network services. Some of the key network configuration files are `/etc/hosts`, `/etc/resolv.conf`, `/etc/host.conf`, and `/etc/sysconfig/network`.

You can work with IPv4 addresses on your LAN. Just assign one of the private IP address ranges for the computers on your LAN. With the right network mask, you can choose from private IP address ranges in Class A, Class B, and Class C. Then all you need is one public IP address to connect your LAN to the Internet. You can use CIDR to configure IP networks with nonstandard network masks.

While broadband connections are often a more cost-effective option for business, most Internet users still connect with a telephone modem. Red Hat has developed an Network Configuration Wizard to help you connect to several types of network adapters, including telephone modems. Alternatively, the `minicom` utility can help you configure an Internet connection from the command-line interface.

When you troubleshoot a network, first remember that most network problems are physical. Check your cables and connections. If that doesn't solve your problems, start collecting data. Work toward identifying the cause of the problem. Research the symptoms.

If none of these approaches helps, step back, take the data you have, and use the scientific method. Linux includes a number of commands that help you collect data and identify the cause of the problem, including `ifconfig`, `netstat`, `ping`, and `traceroute`. The `ifconfig` command helps you make sure that your network adapter is active. The `netstat` command lets you check current network connections and routing tables. The `ping` and `traceroute` commands allow you to check the connectivity within the network.

Now that you know the basics of network configuration, you're ready for Chapter 17, where you'll learn the best practices to secure your network. Red Hat Enterprise Linux includes two key security systems: Pluggable Authentication Modules (PAM) and firewalls.

Chapter 17

Securing Your Linux Network

SECURITY IS IMPORTANT ON any computer network. All types of crackers are out there searching for vulnerable networks. Some look "just for fun," while others break into networks with criminal purposes in mind.

This chapter starts with a general overview of the best practices associated with network security. Some of these practices require good skills with Linux, which you can learn in this book. This chapter covers encryption, firewalls, and passwords, and it addresses the concepts of physical security. Other important skills require good judgment, which may come only with experience.

Red Hat Enterprise Linux requires authentication, not only when users log into their accounts but also when they try to use certain commands or services. The Pluggable Authentication Module (PAM) system is dynamically configurable for any number of situations.

The firewalls you can configure with `iptables` help you customize your system for every service, on every TCP/IP channel. These commands are not difficult to understand, once you know how to break them down into their component parts. And once you understand `iptables`, you can create the firewalls that you need—which will protect you without denying needed services to your users.

Closely related to firewalls is *masquerading*, which hides the true identity of the computers on your LAN from others on the Internet. Masquerading is also a function of `iptables`.

Because no security system is perfect, you'll need to check for break-ins on a regular basis. Tools such as Ethereal let you check what you can see in clear text on the network. You can view log files, such as `wtmp`, to spot unauthorized users. Other tools, such as Tripwire, help you detect changes to critical files.

Yet it is possible to have too much security. If your users aren't following your password policies, those policies may be too difficult. If your users can't get to needed services, perhaps your firewall is too strong. Several other chapters in this book also address detailed requirements for security, from encryption to appropriate configuration of network services. This chapter covers the following topics:

- Understanding best practices
- Using Pluggable Authentication Modules
- Creating firewalls

- Setting up IP masquerading
- Detecting break-ins
- Troubleshooting access issues

Understanding Best Practices

There are a number of steps you can take to secure your network. Some basic practices require more common sense than computer savvy. The way you configure your computers can promote security. Encryption protects data traveling over the network. Good passwords in the right locations protect user accounts and computers. Firewalls also help you provide various degrees of network protection.

Physical Setup

The way you protect your computers and network hardware depends on their value, and on the risks in your environment.

In a home network, it is best to keep hubs and routers out of the reach of toddlers and pets, and in locations where you won't spill coffee. Generally, you aren't worried about people who are trying to physically break into a home network.

In a corporate network, you'll want to secure your computers from sabotage, whether accidental or intentional. Depending on need, you may want to keep your servers, as well as your routers, switches, and hubs, in locked rooms. Secure rooms are also good locations for backup media. Just be sure that these locations have proper environmental controls such as air conditioning to maximize the life of your systems.

TIP *It's important to keep notes on your configuration, just in case you need to reinstall Linux from scratch. Don't keep this file on the same computer, in case you have a hardware failure.*

In a military or other very secure setting, you'll probably be required to take stronger measures, such as removing or locking floppy drives and ports to which you can attach recording hardware. Depending on need, you can configure different levels of physical security for servers, network hardware, and workstations. In addition, you can keep internal networks more secure by isolating them from the Internet.

In any secure setting, consider the use of other basic security systems such as alarms, guards, cameras, ID systems, and similar devices.

Disable Unneeded Services

There are three basic ways to keep a cracker from breaking in through a specific service. You can set up firewalls or other sorts of authentication to keep unauthorized users out. Except for the firewalls discussed in this chapter, most of the associated techniques are specific for each service and are discussed in other chapters.

But it's safer if you can disable or uninstall the service completely. If you disable a service, it's as if you've cut power. However, anyone who breaks in can turn the power back on.

Thus, it's more secure to uninstall a service; that's as if you've removed the wires and motor. But with the magic of RPMs, someone who breaks in may not have direct access to your installation files, so they can't install insecure services on your computer. The commands that we list here are described in Chapters 10 and 13.

Disable

There are two basic things that you need to do to disable a server. You need to turn it off, and you need to make sure it doesn't start automatically the next time your computer boots Linux. For example, the default FTP server is vsFTP. If you want to disable that service, you'll need to use the `service` and `chkconfig` commands. The vsFTP daemon is `vsftpd`; to disable this server, run the following commands.

```
# service vsftpd stop
# chkconfig --level 123456 vsftpd off
```

You can verify that the vsFTP server won't start the next time you boot Linux with the following command:

```
# chkconfig --list vsftpd
```

There are also services associated with the `xinetd` super server. Most of those services are already disabled by default; we'll show you how to disable any active `xinetd` services in Chapter 18.

Uninstall

If you don't need a service, the most secure option is to uninstall the software. For example, if you don't need the vsFTP server, you can uninstall that service with the following command:

```
# rpm -e vsftpd
```

But you may not know the exact name of the service in question. You can get some help from the list of installed RPMs. The `rpm -qa` command lists all currently installed RPMs. That may not be enough, as there are more than 1,000 RPMs that you can install with Red Hat Enterprise Linux. However, you know this is related to the FTP service, so you can identify all related software RPMs with the following command:

```
# rpm -qa | grep ftp
```

This may come up with a whole list of RPMs, including `lftp`, `ftp`, `tftp`, `tftp-server`, and `gftp`. If you're not sure about a particular service, you can find out more. For example, you can get more information about the `tftp-server` RPM with the following command:

```
# rpm -qi tftp-server
```

And alas, a TFTP server is another server that you should normally uninstall or at least disable—unless you actually have one or more diskless workstations on your network. We'll show you how to use a TFTP server for a diskless workstation in Chapter 18.

Encryption

Encrypting sensitive data that you send over a network is a must. In most cases, this means you use a private key to scramble the data you send. On the other end of the connection, you then supply your users with a public key that they use to unscramble your data.

It is possible to activate different levels of security for your passwords, for various services, and for other systems when you installed Red Hat Enterprise Linux. The types of encryption that you can add to your system include the following:

MD5 passwords Linux supports long passwords of up to 256 characters.

Shadow Password Suite This type involves encrypting passwords in `/etc/shadow`, which is normally accessible only to the root user. The suite is active by default (see Chapter 9 for a detailed description).

Kerberos This encryption system eliminates the need to send passwords over a network. With this system, both the client and the server are authorized by a ticket-granting service (TGS). Kerberos is a fully functional encryption system that does not work with the Shadow Password Suite, and is only partially compatible with the PAM system discussed later in this chapter. Kerberos was developed by the Massachusetts Institute of Technology.

GNU Privacy Guard This is commonly used to encrypt e-mail, using the Linux version of the Pretty Good Privacy (PGP) system. GNU Privacy Guard is also used to verify the authenticity of downloads, such as RPMs. See Chapter 10 for more information.

RSA and DSA Digital signature algorithms (DSA) are associated with Secure Shell (SSH) network access. For more information on using SSH with these algorithms, see Chapter 18.

Password Security

At least three levels of password security exist: on the computer, on the bootloader, and when logging into Linux. At each of these levels, you must decide whether you need a password, what type of password you want, and how often you should change that password. Chapter 9 covers the issues and options associated with user passwords.

Passwords on the Computer

Modern PC BIOSs include an option for adding a password for access to the BIOS menu. A BIOS can include a wide variety of options, including a network boot to a computer that may just record passwords that are typed in. Other changes to a BIOS menu could sabotage the data on your system.

However, modifying a BIOS menu, at least on standard PCs, requires physical access to the computer. In other words, if your system is physically secure, you may not need a password on your BIOS.

Passwords on the Bootloader

As we've mentioned before, two basic bootloaders are available: GRUB and LILO. Many users prefer GRUB, because they can protect it with a password. Otherwise, users can change the bootloader configuration file, change the root password by booting Linux in single-user mode, or even access other operating systems, such as Microsoft Windows, that may be accessible in a dual-boot configuration. For

more basic information on GRUB, the default Red Hat Enterprise Linux bootloader, see Chapter 11. Using the techniques discussed in Chapter 11, you can password-protect access to other operating systems. For example, if your computer includes a dual-boot configuration with Microsoft Windows, you can add a password to the appropriate stanza in the GRUB configuration file, `/boot/grub/grub.conf`, as shown here:

```
title DOS
     lock
     password --md5 sf934^(^$asjl
     rootnoverify (hd0,0)
     chainloader +1
```

The `lock` command keeps anyone from booting the associated operating system; attempts result in a `must be authenticated` error message. With this additional code, you first need to enter the password to edit GRUB, select the DOS option, and then enter the MD5 password you created to boot this operating system.

Firewalls and DMZs

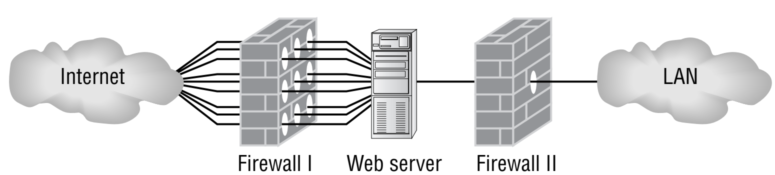
Three basic types of firewalls are available. One can look at every packet of data that comes into your network and make decisions based on the type of data. Another is based on services such as Samba, NFS, and Apache; as we discuss in their respective chapters, various services have their own form of access control that can also serve as a firewall. The third basic type of firewall is based on the services associated with the `xinetd` daemon, as discussed in Chapter 18.

The main Red Hat Enterprise Linux firewall tool is `iptables`. As you'll see later in this chapter, it looks at all data that comes through and allows you to block just the traffic you identify. Alternatively, you can configure it to block all traffic, with exceptions for just the services you need. When you configure a firewall on a gateway computer, it acts as a *bastion host*.

You can set different levels of firewall protection for different computers. For example, if you have a web server, you can configure two different firewalls, as shown in Figure 17.1. For Firewall I, you might configure a minimal level of protection, including commands that help you avoid typical problems associated with web servers, such as the so-called ping of death. For Firewall II, you could include full protection, to help secure your network from the Internet. More information on securing your network from the ping of death and other issues is available later in this chapter.

NOTE *The ping of death is a denial-of-service attack; so much data is sent by a* `ping` *command that no other network messages can get through to the target server. For example, a ping of death to a web server could prevent anyone from getting to any websites installed on that server.*

FIGURE 17.1
Two firewalls



Using Pluggable Authentication Modules

Another level of security is based on Pluggable Authentication Modules (PAM). These modules are typically used to limit access to specific applications, such as `halt` or `redhat-config-network`, to the root user. Different modules let you regulate access by user, password, or access location. Control flags determine whether passing a PAM command line is enough to qualify the user to access the subject application.

NOTE *The definitions associated with PAM often overlap. For the purpose of this chapter, the commands that call PAM modules are* applications*, and commands in PAM module files are* command lines*.*

Basic Configuration

PAM includes a series of dynamically loadable modules that can be customized for specific applications. PAM configuration files are stored in the `/etc/pam.d` directory. Individual modules are stored in the `/lib/security` directory and are documented in the `/usr/share/doc/pam-`*`version`*`/txts` directory.

PAM command lines are all organized in the following format:

```
module_type  control_flag  module_location  arguments
```

Red Hat Enterprise Linux uses PAM modules to secure a substantial number of additional commands. As you can see in Figure 17.2, it includes several basic shell commands as well as almost all the `redhat-config-*` configuration tools.

In the sections that follow, we examine modules and control flags. The module location is simply the location of the file, normally in `/lib/security`. Arguments are associated with each module.

FIGURE 17.2
Red Hat PAM modules

```
[root@Enterprise3 root]# ls /etc/pam.d/
authconfig      printconf-gui                 redhat-config-soundcard
authconfig-gtk  printconf-tui                 redhat-config-time
bindconf        printtool                     redhat-config-users
chfn            reboot                        redhat-config-xfree86
chsh            redhat-cdinstall-helper       redhat-install-packages
cups            redhat-config-authentication  redhat-logviewer
dateconfig      redhat-config-bind            redhat-switch-mail
ethereal        redhat-config-date            redhat-switch-mail-nox
gdm             redhat-config-httpd           rhn_register
gdm-autologin   redhat-config-keyboard        samba
gdmsetup        redhat-config-language        screen
halt            redhat-config-mouse           serviceconf
hwbrowser       redhat-config-netboot         setup
imap            redhat-config-network         smtp.postfix
internet-druid  redhat-config-network-cmd     sshd
kbdrate         redhat-config-network-druid   su
kde             redhat-config-nfs             sudo
kppp            redhat-config-packages        system-auth
login           redhat-config-printer         up2date
neat            redhat-config-printer-gui     up2date-config
other           redhat-config-printer-tui     up2date-nox
passwd          redhat-config-proc            vsftpd
pop             redhat-config-rootpassword    xdm
poweroff        redhat-config-samba           xscreensaver
ppp             redhat-config-securitylevel   xserver
printconf       redhat-config-services
[root@Enterprise3 root]#
```

Module Types

There are four different types of PAM modules, each related to user authentication:

Password Linux login consoles don't allow users to try to log in again and again, at least not easily. This is because of a PAM password module that sets limits for the number of attempted logins and password length.

Session This type of module creates settings for an application. For example, PAM session modules can limit the number of times any specific user can log into a Linux server.

Account This type of module manages access based on policies. For example, PAM account modules can allow or deny access based on a user list, time, or password expiration.

Auth Short for *authentication*, an auth module checks the identity of a user. For example, PAM authentication modules can prompt for a username and password.

A common argument for each module is `service=system-auth`, which calls the `system-auth` PAM module for username and password requirements.

Control Flags

There are four possible control flags for each PAM command line. These flags, shown in Table 17.1, determine the action of the application when the module command succeeds or fails.

TABLE 17.1: CONTROL FLAGS IN PAM

CONTROL FLAG	DESCRIPTION
`optional`	The module doesn't really matter, unless all other modules also have the `optional` control flag.
`required`	If the module fails, the application associated with this file also fails.
`requisite`	If the module fails, immediately stop the authentication process and don't allow use of the command; later commands in the PAM file are ignored.
`sufficient`	If the module succeeds, immediately stop the authentication process, and OK the use of the command; later commands in the PAM file are ignored.

A PAM Example

To understand how PAM modules work, it is helpful to analyze a PAM configuration file, line by line. All PAM configuration files are located by default in `/etc/pam.d`. The following code example is based on the `redhat-config-xfree86` file in this directory. You'll see that this configuration file has the same name as the configuration utility discussed in Chapter 29. Let's take this file, line by line.

```
auth  sufficient  pam_rootok.so
```

The `auth` module type tells you this command line is going to check the identity of a user. The `sufficient` control flag lets the application run if this command line succeeds. The `pam_rootok.so` module in the `/lib/security` directory returns `PAM_SUCCESS` if the user is root. In other words, if the root user runs `redhat-config-xfree86`, no other command lines in this file are run, and the application starts.

```
auth  sufficient  pam_timestamp.so
```

This command also uses the auth module type with a `sufficient` control flag. The `pam_timestamp.so` module normally returns `PAM_SUCCESS` for regular users who have run `sudo` in the past five (5) minutes.

```
auth   required   pam_stack.so service=system-auth
```

This command uses the auth module type with a `required` control flag. The `pam_stack.so` module returns `PAM_SUCCESS` if the `service=system-auth` argument is satisfied. The `system-auth` module requires the user to enter the root password.

```
session  required  pam_permit.so
```

This command uses the session module type with a `required` control flag. The `pam_permit.so` module always returns `PAM_SUCCESS`, so proceed to the next line.

```
session  optional  pam_xauth.so
```

This command uses the session module type with an `optional` control flag. The `pam_xauth.so` module does not return success or failure. The `optional` flag makes this command line trivial with respect to this file. However, you can add a `debug` argument to log access requests in `/var/log/messages`.

```
session  optional  pam_timestamp.so
```

This command also uses the session module type with an `optional` control flag. The `pam_timestamp.so` module updates any available time stamp file, normally located in the `/var/run/sudo` directory. There's one more command in this file.

```
account  required  pam_permit.so
```

This command uses the account module type with a `required` control flag. The `pam_permit.so` module always returns `PAM_SUCCESS`.

Creating Firewalls

Any command or configuration file that is configured to block data from coming into your system or LAN is a *firewall*. Some of these commands and configuration files are covered in other chapters. The main Linux firewall tool is `iptables`. Various `iptables` commands can be connected in chains. Each of these commands can be used to block or allow data associated with specific protocols.

Other Firewall Commands

The two legacy alternatives to `iptables` are `ipfwadm` and `ipchains`. The `ipfwadm` command is associated with the Linux kernel 2.0.x and is now obsolete. The `ipchains` command is associated with the Linux kernel 2.2.x and is still supported in the current Linux 2.4.x kernel.

While there are many secure `ipchains` firewalls, this command is not supported in Red Hat Enterprise Linux 3, and the associated `ipchains` RPM is not included with this distribution.

Data Directions and *iptables*

The `iptables` command is based on regulating data traffic in three directions: in, out, and through. In other words, you can configure `iptables` to stop data from coming in from an outside network. You can configure `iptables` to stop data from leaving your computer. And you can configure `iptables` to regulate data that travels forward through your computer—that is, between a LAN and another network such as the Internet.

IPv6 Firewalls

Red Hat Enterprise Linux also includes a firewall tool for those of you who configure networking using IPv6 addresses. Naturally, the tool is `ip6tables`. The format and syntax of that command is the same as for `iptables`. But if you want to run `ip6tables`, you'll need to deactivate the `iptables` service first.

To deactivate `iptables`, you'll need to run the following commands. The first command turns off the service; the second command makes sure it does not start the next time you boot Linux:

```
# service iptables stop

# chkconfig --level 2345 iptables off
```

Now you can activate the `ip6tables` service. The corresponding commands are straightforward.

```
# service ip6tables start

# chkconfig --level 2345 ip6tables on
```

Firewalls as Chains

No magic `iptables` command is available that works for everyone. Most firewalls are based on a series of `iptables` commands that are connected as chains. Let's take a look at a fairly simple firewall, based on a high-security firewall created during the installation of Red Hat Enterprise Linux. The entries shown in Figure 17.3 are from `/etc/sysconfig/iptables`, where Red Hat Enterprise Linux saves firewall commands.

For the moment, just note that four different chains are shown in this file: `INPUT`, `FORWARD`, `OUTPUT`, and `RH-Firewall-1-INPUT`. The first three chains are default chains that allow all traffic to flow through the firewall. All of the commands that follow the `-A` are appended to the end of the RH-Firewall chain. In the following sections, we explain `iptables` commands and options in more detail.

FIGURE 17.3
An *iptables* firewall

```
# Firewall configuration written by redhat-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -i eth1 -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
~
~
"/etc/sysconfig/iptables" 22L, 1041C
```

Format of *iptables*

Let's analyze the `iptables` command in detail. This is a rich command; entire books are available that explore the various associated options. While we describe the masquerading options later in this chapter, let's look at a few important options now. The `iptables` command has a very specific format:

```
iptables -t table option pattern -j target
```

The first option here is based on the `-t` *`table`* option. Two basic tables are available: `filter` and `nat`. The `nat` table supports the Network Address Translation associated with masquerading. The `filter` table allows you to block or allow specific types of network traffic. Because `-t filter` is the default, this option is usually not specified in a firewall configuration file.

Options for *iptables*

Remember, there are three default chains: `INPUT`, `OUTPUT`, and `FORWARD`. Four main options are associated with `iptables`: you can list (`-L`), append (`-A`), or delete (`-D`) a specific rule, or flush (`-F`) all of the rules in a chain.

The `iptables -L` command lists all of the current rules on all chains. If your firewall is complex, you may want to list the rules on a specific chain. For example, the `iptables -L INPUT` command lists all firewall rules related to data coming into your computer. A sample list of current firewall rules is shown in Figure 17.4.

To add a new rule, you'll generally append it to the end of one of the chains. For example, the following command appends a limit of a packet every second to the `ping` command to data that is forwarded through your computer, thus preventing the so-called ping of death:

```
# iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
```

To delete an existing rule, first identify the chain and the location of the rule within the chain. For example, if you want to delete the rule related to accepting web (`http`) requests in Figure 17.4, note that it's the seventh rule in the `RH-Firewall-1-INPUT` chain. The appropriate command is

```
# iptables -D RH-Firewall-1-INPUT 7
```

FIGURE 17.4
Current *iptables* rules

```
[root@Enterprise3 root]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere           icmp any
ACCEPT     ipv6-crypt--  anywhere             anywhere
ACCEPT     ipv6-auth--  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere           state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:smtp
REJECT     all  --  anywhere             anywhere           reject-with icmp-host-prohibited
[root@Enterprise3 root]#
```

If you're a bit frustrated, you can start over. For example, if you had a series of rules in the `FORWARD` chain that you wanted to delete, run the following command:

```
# iptables -F FORWARD
```

This command can be a bit dangerous; if you ran the `iptables -F` command without specifying a chain, you would delete every rule in every chain. Basic `iptables` options are shown in Table 17.2.

TIP *If you accidentally flush your* `iptables` *chains, the original chains should still be available in* `/etc/sysconfig/iptables`*. You can make Linux reread these rules with the* `service iptables reload` *command.*

TABLE 17.2: OPTIONS FOR *IPTABLES*

OPTION	**FUNCTION**
-A *chain rule*	Appends a rule to the end of a *chain*
-D *chain number*	Deletes the rule number from the specified *chain*
-F *chain*	Flushes, or deletes, all rules from the specified *chain*
-I *chain number rule*	Inserts a rule as the specified rule number in the noted *chain*
-L *chain*	Lists the current rules in the specified *chain*
-N *chain*	Starts a new nonstandard *chain*
-X *chain*	Deletes a user-defined *chain*

Patterns for *iptables*

Now it's time to examine the next step in the `iptables` command. Previously, you've identified the action to take on a chain. Next, you need to specify a pattern to match in the chain. Patterns can match the IP address of the message sender or source, the TCP/IP port, and or the protocol.

IP Address Patterns

Take the previous command that prevents the ping of death. For some reason, say you want to regulate the `ping` command solely from IP address 199.88.77.66. You could do so with the following command:

```
# iptables -A FORWARD -s 199.88.77.66 -p icmp --icmp-type
➥echo-request -m limit --limit 1/s -j ACCEPT
```

Note the use of the `-s` option, which prepares the way for the source IP address. You could reverse the effect and regulate the `ping` command from every other address, by using an exclamation point:

```
# iptables -A FORWARD -s !199.88.77.66 -p icmp --icmp-type
➥echo-request -m limit --limit 1/s -j ACCEPT
```

The exclamation point (`!`) tells `iptables` to treat whatever follows as an exception. In other words, this command is applied to every computer on the Internet unless it has the noted IP address.

It helps to specify a range of IP addresses such as a LAN. The following commands combine a network IP address with a subnet mask in regular and CIDR notation. (See Chapter 16 for a description of CIDR, which is short for Classless Inter-Domain Routing.)

```
# iptables -A FORWARD -s 199.88.77.0/255.255.255.0 -p
å icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
# iptables -A FORWARD -s 199.88.77.0/24 -p icmp --icmp-type
å echo-request -m limit --limit 1/s -j ACCEPT
```

Some of the other switches associated with `iptables` are shown in Table 17.3.

Table 17.3: Switches for *iptables*

Switch	Function
`--dport` *port*	Specifies the destination TCP/IP port number.
`--icmp-type` *message*	Allows you to specify the type of ICMP message; `echo-request` corresponds to the messages sent by a `ping` command.
`-j` *action*	Notes an action to be taken if the requirements of the command are satisfied—normally `ACCEPT`, `DROP`, `REJECT`, or `LOG`.
`--limit` *time*	Sets an allowable rate for a specific message; can be in seconds, minutes, hours, or days; e.g., `2/s` = 2 per second.
`-m` *condition*	Looks at the data for a match; may be a protocol, such as `tcp` or `udp`, or a condition, such as a limit.
`-p` *protocol*	Checks the data for a specific protocol, such as `tcp` or `udp`.
`-s` *ip_address*	Specifies a source IP address.
`--sport` *port*	Sets a source TCP/IP port.
`--tcp-flags` *fl1,…*	Looks for flags in a TCP packet:

Continued on next page

TABLE 17.3: SWITCHES FOR *IPTABLES (continued)*

SWITCH	FUNCTION
`--tcp-flags fl1,...` (cont.)	SYN (synchronize) packets are sent from a client and expect a reply.
	ACK (acknowledgment) packets acknowledge SYN requests.
	A FIN (finish) packet is the final one in a communication.
	RST (reset) packets tell a client that a request has been rejected.
	Example: `--tcp-flags SYN,RST,ACK SYN` looks for SYN, RST, and ACK packets but passes only packets that have the SYN flag.

TCP/IP PROTOCOL PATTERNS

The `iptables` command looks at every data packet that comes in, goes out, or forwards through your computer. You can tell the command to look for a specific protocol. The most common protocol patterns are based on TCP, UDP, and ICMP. The key is the `-p` option, which specifies the protocol. For example, the earlier command that prevents the ping of death uses the `-p icmp` option, since `ping` is associated with ICMP. (For more information on ICMP, see Chapter 15.)

TCP/IP PORT PATTERNS

As noted in Chapter 15, over 65,000 TCP/IP ports are available. Many of these ports are dedicated to standard services. For example, the following command stops any attempt to connect from the 199.88.77.0/24 network with TCP packets to port 21, which is associated with FTP:

```
# iptables -A FORWARD -s 199.88.77.0/24 -p tcp --dport 21 -j REJECT
```

Actions for *iptables*

Say you've created an `iptables` command that looks for some pattern in the data that goes into, out of, or through your computer. But if it finds a match, you need to tell `iptables` what to do with that packet of data.

When `iptables` finds a match, the `-j` command tells the chain to jump to one of four conclusions: `ACCEPT`, `DROP`, `REJECT`, or `LOG`. These actions are explained in Table 17.4.

TABLE 17.4: ACTIONS FOR *IPTABLES*

ACTION	EXPLANATION
`-j ACCEPT`	Allows packets that match the specified characteristics into, out of, or through your computer.
`-j DROP`	Stops packets that match the specified characteristics into, out of, or through your computer.
`-j REJECT`	Stops packets that match the specified characteristics into, out of, or through your computer; a message is sent to the computer that sent the message.
`-j LOG`	Logs a record of matching packets in `/var/log/messages`.

Putting It All Together

Now that we've broken down the `iptables` command, you can create the firewall rules that you need. While tools such as `redhat-config-securitylevel` can help, GUI tools do not give you the degree of control that you may need. You need to know at least how to add and delete rules from a firewall chain.

Starting without a Firewall

As an experiment, let's start with a computer without a firewall. This assumes you have a LAN of two or more computers. If you have firewall rules in `/etc/sysconfig/iptables` that you want to save, back them up. Append the rule discussed earlier on the ping of death. Revise it so it drops any ping requests from within your LAN.

The following steps assume a LAN with an address of 192.168.0.0/24; if your LAN has a different address and network mask, substitute accordingly.

1. Back up any current firewall. Copy `/etc/sysconfig/iptables` to a file in your home directory.
2. Flush any rules in your current firewall with the `iptables -F` command.
3. Append the ping of death rule as shown. This stops any pings to your computer (`INPUT`) from the cited network:

   ```
   # iptables -A INPUT -s 192.168.0.0/24 -p icmp --icmp-type echo-request -j DROP
   ```

4. Try the `ping 127.0.0.1` command on the local computer. It should still work.
5. Go to another computer on your LAN. Try to `ping` the IP address of the first computer. You should see a one-line response before everything stops.
6. If necessary, restore the original `/etc/sysconfig/iptables` file.

If you're in a mood for experiments, try these steps again, this time with a `-j REJECT` option at the end of the `iptables` command. Note the difference when you run the `ping` command from the other computer on your LAN.

Inserting a Firewall Rule

Return to the firewall described earlier, depicted in Figure 17.4. If you install a web server on your computer in the future, you'll want to revise your firewall a bit. The current firewall includes rules as shown by an `iptables -L` command:

```
Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source        destination
ACCEPT     all  --  anywhere      anywhere
ACCEPT     all  --  anywhere      anywhere
ACCEPT     icmp --  anywhere      anywhere       icmp any
ACCEPT     ipv6-crypt--  anywhere    anywhere
ACCEPT     ipv6-auth--  anywhere    anywhere
```

```
ACCEPT    all  --  anywhere     anywhere      state RELATED,ESTABLISHED
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:http
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:ftp
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:ssh
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:telnet
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:smtp
REJECT    all  --  anywhere     anywhere      reject-with icmp-host-prohibited
```

You need to insert an `iptables` rule that accepts secure web data through TCP/IP port 443. Based on the conditions described earlier:

- We're inserting a rule in the chain named `RH-Firewall-1-INPUT`. Make it the eighth rule in the chain (`-I RH-Firewall-1-INPUT 8`).
- Since connections to a website need a reply, they require TCP packets (`-p tcp`).
- We know from `/etc/services` that connections to a secure website work through port 443 (`-m tcp --dport 443`).
- Requests to secure websites come from clients and should have `SYN` flags. They should be checked for `RST` and `ACK` flags to make sure they're not coming from other computers acting as servers (`--tcp-flags SYN,RST,ACK SYN`).
- Finally, packets that meet all of these conditions should be accepted (`-j ACCEPT`).

Putting this all together, we end up with the following command:

```
# iptables -I RH-Firewall-1-INPUT 8 -p tcp -m tcp
➥ --dport 443 --tcp-flags SYN,RST,ACK SYN -j ACCEPT
```

Once you add the command, you can see the following result in the `iptables` chain:

```
Chain RH-Firewall-1-INPUT (2 references)
target    prot opt source       destination
ACCEPT    all  --  anywhere     anywhere
ACCEPT    all  --  anywhere     anywhere
ACCEPT    icmp --  anywhere     anywhere      icmp any
ACCEPT    ipv6-crypt--  anywhere    anywhere
ACCEPT    ipv6-auth--  anywhere    anywhere
ACCEPT    all  --  anywhere     anywhere      state RELATED,ESTABLISHED
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:http
ACCEPT    tcp  --  anywhere     anywhere      tcp dpt:https flags:SYN,RST,ACK/SYN
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:ftp
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:ssh
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:telnet
ACCEPT    tcp  --  anywhere     anywhere      state NEW tcp dpt:smtp
REJECT    all  --  anywhere     anywhere      reject-with icmp-host-prohibited
```

Note how `iptables` has converted the port number (443) to the associated protocol (`https`). If this is what you want to do, remember to save your configuration changes.

Saving Configuration Changes

You can save configuration changes to `/etc/sysconfig/iptables` with the `service iptables save` command.

While `iptables` is the default for Red Hat Enterprise Linux 3, it is always a good idea to check the service status of your firewall. You can do so with the `chkconfig` command. For example, the following command should show the runlevels where Linux starts the `iptables` service:

```
# chkconfig --list iptables
iptables   0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

If you see that the `iptables` service is not set to activate (and at the right runlevels), you can make it happen. For example, the following command activates `iptables` the next time you start in runlevel 2, 3, or 5:

```
# chkconfig --level 235 iptables
```

NOTE *Remember, Red Hat Enterprise Linux does not normally use runlevel 4. For details, see Chapter 11.*

The Red Hat Security Level Tool

The Red Hat Firewall configuration tool is `redhat-config-securitylevel`, which is essentially the same tool that you used during the installation process in Chapter 3 or 4. Alternatively, you can start it from a GNOME desktop by selecting Main Menu ➢ System Settings ➢ Security Level. This opens the Security Level Configuration window, shown in Figure 17.5.

NOTE *Before you make changes using the Red Hat tools, back up your current firewall settings. As noted earlier, they're stored in the* `/etc/sysconfig/iptables` *file.*

Unlike older Red Hat distributions, this tool only allows you to activate or deactivate a firewall. If you choose to activate a firewall, you can customize the traffic that it blocks.

FIGURE 17.5
Setting up a firewall

For example, if one of the network cards is connected only to the local network, you may want it to be a trusted device; firewall rules do not apply to traffic through trusted devices. In Figure 17.5, `eth0` is a trusted device, and traffic that comes in through that network card is not affected by the firewall. In addition, you can customize the firewall to allow incoming data associated with the protocols shown in the Security Level Configuration window.

If you're using the default `iptables` firewall command, any changes that you make are written to `/etc/sysconfig/iptables`.

The Console Security Level Tool

In this case, the console-based version of the Red Hat firewall tool is more flexible than `redhat-config-securitylevel`. You can start this console tool with the `redhat-config-securitylevel-tui` command. This opens the menu shown in Figure 17.6.

FIGURE 17.6
The *flexible redhat-config-securitylevel-tui* tool

The `redhat-config-securitylevel-tui` tool is the successor to `lokkit`. For now, the `lokkit` command still works, but Red Hat is in the process of changing the commands that start almost all of its tools to the `redhat-config-*` format.

As you can see, you can Enable or Disable the firewall. For this exercise, Enable the firewall and select Customize. You're taken to the Firewall Configuration - Customize menu shown in Figure 17.7 where you can allow incoming data for the same standard services as the GUI tool. You can also enable access through other ports, such as the secure HTTP service.

In the figure, I've enabled communication through the ports required for Samba connections. You can see the results in Figure 17.8, which is the output from an `iptables -L` command. For more information on most TCP/IP port numbers, see the `/etc/services` file. The list is not 100 percent complete; the official list is kept in `www.iana.org/assignments/port-numbers`.

FIGURE 17.7
Customizing the firewall

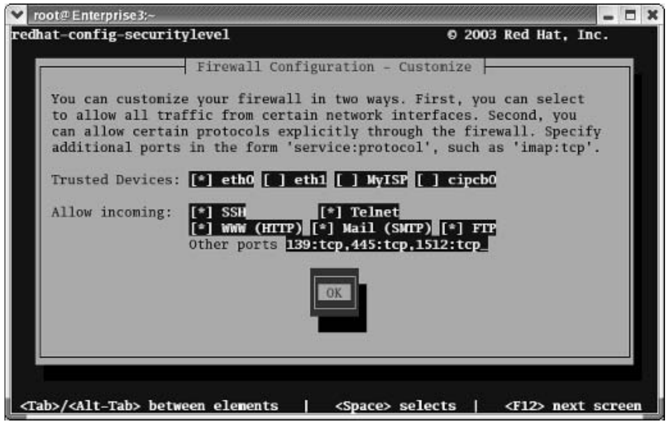


FIGURE 17.8
List of firewall rules

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere           icmp any
ACCEPT     ipv6-crypt--  anywhere             anywhere
ACCEPT     ipv6-auth--  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere           state RELATED,ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere           state NEW udp dpt:135
ACCEPT     udp  --  anywhere             anywhere           state NEW udp dpt:netbios-ns
ACCEPT     udp  --  anywhere             anywhere           state NEW udp dpt:netbios-dgm
ACCEPT     udp  --  anywhere             anywhere           state NEW udp dpt:wins
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:135
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:netbios-ssn
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:microsoft-ds
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:wins
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:smtp
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere           state NEW tcp dpt:ftp
REJECT     all  --  anywhere             anywhere           reject-with icmp-host-prohibited

[root@Enterprise3 root]#
```

Rebuilding a Firewall

If you make changes to a firewall and make mistakes, you can start over again. If you've made changes directly, by running `iptables` from the command-line interface, you can restore your original firewall by restarting the service.

```
# service iptables restart
```

If you've changed your firewall using one of the Red Hat tools, you can restore your original firewall settings to `/etc/sysconfig/iptables`. You can then restart the `iptables` service.

Setting Up IP Masquerading

IP masquerading allows you to hide the IP addresses of the computers on your LAN. It replaces these IP addresses with the public IP address on your gateway computer. This helps to protect the computers within your LAN from direct attack.

NOTE *IP masquerading is a form of Network Address Translation (NAT). Another way to implement NAT is with a proxy server.*

IP masquerading and firewalls are commonly configured on the same computer on a LAN, most commonly the gateway between that LAN and an external network such as the Internet. Therefore, the developers of `iptables` have included options to use that command to configure masquerading.

Naturally, a gateway computer for a LAN serves as one that routes messages between networks. To make this work, you need to enable IP Forwarding, as described in Chapter 16.

You can't configure masquerading using the Red Hat firewall tools. Once you've set it up, you'll want to store the command. The best place is `/etc/sysconfig/iptables`, which is read and run during the Red Hat Enterprise Linux 3 boot process.

Functionality

As described in Chapter 16, you can configure a gateway computer to connect to your LAN and another network such as the Internet. Assuming that you're connecting to the Internet, you can use private IP addresses within your LAN and use a public IP address on the network card that is connected to the Internet.

Then to complete the connection, you must configure IP Forwarding on the gateway computer as described in Chapter 16. And then, you need to add an appropriate `iptables` command to your firewall.

Once you've set up masquerading, anyone who connects to the Internet from inside your LAN sends data packets through your gateway computer. For example, assume one of your users is looking for a website. The source address—that is, the IP address of the computer on your LAN—is replaced with the public IP address of your gateway computer. The `iptables` command assigns a nonstandard TCP/IP port to the packet. The gateway computer then caches the source IP address and the assigned TCP/IP port.

When the firewall receives the data for the website, the process is reversed. The assigned port is matched to the cache. The IP address of the source computer is taken from the cache and added to the data for the website. The gateway computer can then send the packets to the source computer.

IP Masquerading Commands

Let's take another look at the format of the `iptables` command. As discussed earlier, the default table is a filter, which is the firewall function associated with `iptables`.

```
iptables -t table option pattern -j target
```

However, a `-t nat` option is available that allows you to use `iptables` to configure masquerading. For example, the following command assumes that your network has an address of 10.0.0.0/24 and that the network card on your gateway that's directly connected to the Internet is `eth2`.

```
# iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth2 -j MASQUERADE
```

This command changes the IP address of the packets that are going out to the Internet (`-A POSTROUTING`), and the changes are only good for the private IP addresses on your LAN (`-j MASQUERADE`).

Detecting Break-ins

There are two standard ways to see if a cracker has broken into your system. One is to check logins as documented in the `/var/log/wtmp` file. The other is to check log file activity to see when the traffic on your Linux systems should be at a minimum.

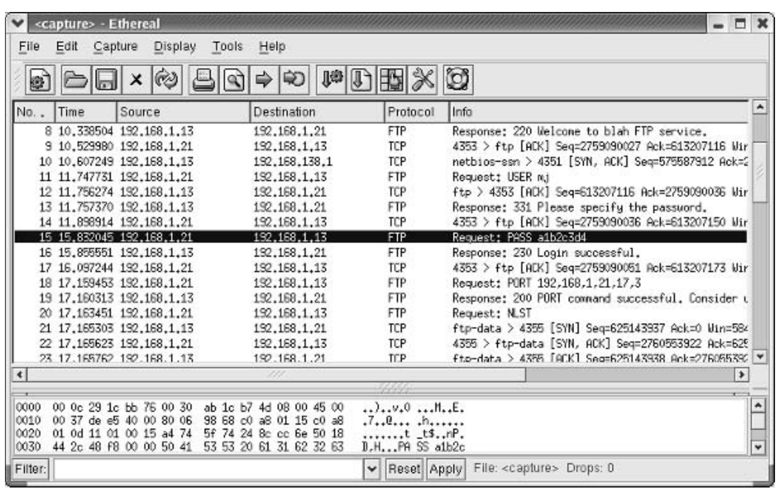
But one of the ways people break into a system is by reading the clear-text passwords that a user may send over the network. One useful tool for looking at network traffic is Ethereal, a protocol analyzer that is available for Linux/Unix and Microsoft Windows. It's included with Red Hat Enterprise Linux in the `ethereal-*` RPM packages.

Sniffing with Ethereal

A more descriptive but colloquial name for a protocol analyzer is a *sniffer*. Protocol analyzers such as Ethereal record, or "sniff," the traffic on a network. If you're on an Ethernet network, you can record all communication between all computers on the LAN.

If a message is transmitted in clear text, Ethereal converts it into a readable format. For example, take Figure 17.9, which shows an Ethereal view of various network packets. Note the highlighted packet number 15 carefully.

FIGURE 17.9
Ethereal reveals a password.



As you can see, packet 15 shows the password that user mj (see packet 11) entered to connect to the local FTP server: `a1b2c3d4`.

This illustrates one reason why physical security on a network is so important: if crackers can gain physical access to a LAN, they can connect a computer with Ethereal and find the password of anyone who uses a clear-text server on that LAN.

Ethereal is far from the most sophisticated tool that a cracker can use. If you can detect a clear-text password with Ethereal, you know that a cracker could read that password as well.

Once you have installed the `ethereal-*` RPM packages, you can start this tool with the `ethereal` command.

Checking Logins

It's a good idea to inspect your log files for suspicious activity. For example, login records are available in the `/var/log/wtmp` file. Because this is a binary file, you need a binary reader, `utmpdump`, for this purpose. Read the records of this file by issuing the `utmpdump /var/log/wtmp` command. An excerpt from my output is shown in Figure 17.10.

Note the second-to-last entry in Figure 17.10. As you can see, the originating IP address is 128.99.1.64. If that does not belong to an authorized computer or network, you should be concerned. Someone may be trying to break into your system. You may then consider adding `iptables` firewall commands that would block access from this IP address or the associated IP network.

Tripwire and Suspicious Activity

You learned about how log files are configured through `/etc/syslog.conf` in Chapter 13. Most log files are stored in the `/var/log` directory; log entries are stamped with a time of day. You can view different log files periodically to check for suspicious activity at times when there should be no activity on your system or your network.

Unfortunately, a skilled cracker will try to fool you into believing that everything is all right on your system. For example, a cracker with root access could replace the files in your `/var/log` directory.

FIGURE 17.10
Reviewing login activity

```
[5] [01742] [4   ] [        ] [            ] [2.4.21-4.EL         ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[5] [01743] [5   ] [        ] [            ] [2.4.21-4.EL         ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[5] [01744] [6   ] [        ] [            ] [2.4.21-4.EL         ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[6] [01741] [3   ] [LOGIN   ] [tty3        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[6] [01742] [4   ] [LOGIN   ] [tty4        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[6] [01744] [6   ] [LOGIN   ] [tty6        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[6] [01739] [1   ] [LOGIN   ] [tty1        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[6] [01740] [2   ] [LOGIN   ] [tty2        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[6] [01743] [5   ] [LOGIN   ] [tty5        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:52:49 2004 EDT]
[7] [01739] [1   ] [root    ] [tty1        ] [                    ] [0.0.0.0
    ] [Mon Apr 05 11:53:13 2004 EDT]
[7] [01900] [/0  ] [root    ] [pts/0       ] [:0.0                ] [128.99.1.64
    ] [Mon Apr 05 12:08:48 2004 EDT]
[7] [01900] [/1  ] [root    ] [pts/1       ] [:0.0                ] [0.0.0.0
:
```

One important tool for checking the integrity of your files is Tripwire. As of this writing, there is both an open-source and a commercial version of this software. Unfortunately, the open-source version is not included with the Red Hat Enterprise Linux Installation RPMs. You can download and read the documentation at `www.tripwire.org`; the commercial version is available as part of the TriSentry suite from Psionic Technologies (`www.psionic.com`).

Tripwire is designed to check the integrity of key configuration files on your system. For the tool to be effective, you should install it as soon as possible; it can't detect unwanted changes after a cracker has broken into and changed key files on your system.

Note *Tripwire is not included with Red Hat Enterprise Linux and is not supported by Red Hat.*

Since Tripwire is not included with Red Hat Enterprise Linux, you'll need to install it from another source. I've installed it from the Red Hat Linux 9 RPM packages. The same basic package is available for Fedora core, so I suspect it may be included with Red Hat Enterprise Linux again in the future.

Once you've installed Tripwire, you need to set it up and create a basic database. Then the `cron` job that comes with the Tripwire RPM can check your files on a daily basis (oddly enough, the `cron` job is named `tripwire-check`).

Setting Up Tripwire

It's easy to set up Tripwire. Just run the installation script, `/etc/tripwire/twinstall.sh`. The script is in text format; you can even use a text editor to modify the locations of installation files. It includes a copy of the Tripwire license, the GPL.

When you run the default script, you're prompted to add local and site *passphrases*, which are passwords used to encrypt access to Tripwire. During the setup process, the `twinstall.sh` script also creates a configuration and policy file in the `/etc/tripwire` directory.

Next, initialize the Tripwire database with the `tripwire --init` command. This command may take a few minutes as it uses its policy file, `tw.pol`, to build an initial database. It may cite a few errors as it searches for files that you may not have installed.

You can update the Tripwire policy file by editing `/etc/tripwire/twpol.txt`. For example, if you haven't installed the "Z" shell, you could delete the reference to `/bin/zsh`. Once your changes are complete, you can update Tripwire policies with the following command:

```
# tripwire --update-policy /etc/tripwire/twpol.txt
```

Tip *Once you install it, Tripwire is an important tool for defending your system. A cracker may try to hide his or her tracks by changing various tripwire files. You can prevent this by using some secure or read-only media; for example, some administrators write Tripwire files to a read-only CD.*

Tripwire in Action

Assuming you installed Tripwire, the database is checked daily. In fact, there is a `tripwire-check` script in the `/etc/cron.daily` directory. As discussed in Chapter 13, this script is run by default, at 4:02 a.m. every morning, through `/etc/crontab`.

You may want to edit this file to save the output; for example, you may direct the output from the `tripwire` command to a log file:

```
/usr/sbin/tripwire --check >> /var/log/tripwire
```

The resulting output is interesting. For the purpose of this book, I temporarily changed the name of the `/sbin/halt` file before running the `tripwire-check` script. In `/var/log/tripwire`, this led to a lot of output, including the following lines:

```
--------------------------------------------------------
Rule Name: User binaries (/sbin)
Security Level: 66
--------------------------------------------------------

Added:
"/sbin/halt.bak"

Modified:
"/sbin"
--------------------------------------------------------
Rule Name: System Administration Programs(/sbin/halt)
Security Level: 100
--------------------------------------------------------

Removed:
"/sbin/halt"
```

While this warning seems subtle, it tells you that someone has deleted the `halt` command from your Linux system. As you can deduce from the first few lines, I actually renamed the `/sbin/halt` file to `/sbin/halt.bak`.

NOTE *It isn't quite this simple for a Tripwire package from Red Hat Linux 9. There are a number of other files that are different from Red Hat Enterprise Linux 3. However, you can at least set a baseline log and detect problems based on differences with this log.*

Troubleshooting Access Issues

It is possible to have too much security. Any security measure that is keeping users from needed services is probably doing more harm than good.

If your users need a service and your security measures block the use of that service, you need to make a choice. You can either provide an acceptable substitute or you can relax your security measures in some way.

Sometimes, users may tell you that something is not working when it is really an issue with your security. For example, the `iptables DROP` option can lead to output that is confusing to users.

Too Much Security

Security is not helpful if it keeps users from getting their work done. However, some services are sufficiently dangerous that you need to provide your users with alternatives.

For example, if a user wants to connect to a remote computer via Telnet, it's probably in your best interest to help that user learn about the Secure Shell utilities described in Chapter 18. You generally don't want users sending their user passwords in clear text over the network.

Another example is with the Network File System (NFS), which is detailed in Chapter 22. NFS requires access to several different TCP/IP services: `nfs`, `portmap`, `rpc.mountd`, and `rpc.nfsd`. While NFS uses TCP/IP port 2049, standard Red Hat Enterprise Linux firewalls also block port 111, which is associated with the RPC daemon.

Denial or Rejection

When users try to access a prohibited service, what they see depends on `iptables`, specifically the `DROP` or the `REJECT` option. For example, you could implement either of the following chains to stop users on the 192.168.0.0/24 network from connecting via the Secure Shell (SSH):

```
# iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 22 -j DROP
# iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 22 -j REJECT
```

TIP *Port 22 is the TCP/IP port for the SSH service. You can look up standard TCP/IP ports in* `/etc/services`*.*

Now compare what a user on the 192.168.1.0/24 network sees when she tries to connect via Telnet to a server that is set to `DROP` a request.

```
# telnet Enterprise3
Trying 192.168.1.13...
```

A user who sees this output may complain to you that Telnet is not working. Now contrast that with the output for someone who is trying to connect to a server that is set to `REJECT` a request:

```
# telnet Enterprise3
Trying 192.168.1.13...
telnet: Unable to connect to remote host: Connection refused
```

A user who sees this is more likely to understand that Telnet connections aren't allowed on server Enterprise3. When the user asks you why, you have an opportunity to educate your user about alternatives such as SSH.

NOTE *The Telnet service included with Red Hat Enterprise Linux 3 is a Kerberos-enabled version of this otherwise insecure service. Nevertheless, SSH is generally the preferred option. We describe both services in Chapter 18.*

Summary

You may be lucky. You may be administering a network that isn't connected to any other network, especially the Internet. Your servers and network components could be secure in locked rooms. And you may be able to trust your users. In this case, you may not need to secure your Linux network.

However, most LANs are connected to other networks. Many users need Internet connections to be productive. Unfortunately, any Internet access can expose your LAN to crackers who want to break into your systems.

There are best practices associated with network security, such as providing various levels of physical security for your computers and network components: configuring different levels of firewalls for your web server and internal LAN, encrypting communications with various protocols such as Kerberos and GPG, encrypting your passwords using MD5 and shadow passwords, and providing different levels of password security on your BIOS and Linux bootloader.

Pluggable Authentication Modules (PAM) let you limit access to specific applications, as defined in the `/etc/pam.d` directory. The files in this directory are associated with different applications. The four types of PAM modules are password, session, account, and auth. Each module is associated with one of four control flags: `optional`, `required`, `requisite`, and `sufficient`. These control flags drive the response to the module.

The main Red Hat Enterprise Linux firewall utility is `iptables`. Various `iptables` commands can be connected in chains for data in three directions: `INPUT`, `OUTPUT`, and `FORWARD`. You can configure `iptables` to match different patterns: IP addresses, TCP/IP ports, even patterns that can prevent the ping of death. When a firewall command matches a pattern, you can set `iptables` to `ACCEPT`, `DROP`, `REJECT`, or `LOG` the occurrence. The Red Hat GUI and console `redhat-config-securitylevel*` tools can help you configure your firewall.

You can also configure `iptables` for IP Masquerading. This is a form of Network Address Translation that hides the address of the computers on your LAN requesting access to an outside network such as the Internet. Each outgoing packet is associated with an unused port number; when the LAN gets an answer, that number is used to identify the requesting computer.

There are a number of ways to detect attempted break-ins to your Linux computer. One is to check logins to `/var/log/wtmp`. Another is to use the Tripwire RPM package. It's also useful to check your traffic with Ethereal; it tells you if users are sending their passwords over the network in clear text.

Of course, it is possible to have too much security. Any measure that keeps your users from needed services may be too strong. The way you configure `iptables` can confuse your users.

In the next chapter, we'll examine other ways to access computers through the network. Some are not secure such as the Remote Shell and Telnet. On the other hand, the Secure Shell is quite secure, because it encrypts communication with passphrases and more. You can also help protect even insecure services using the `tcp_wrappers` access control files.

Part 5

Basic Linux Services

In this part, you will learn:

- **Chapter 18: Remote Environments**
- **Chapter 19: DNS and DHCP**
- **Chapter 20: Printing with CUPS**
- **Chapter 21: Mail Services**

Chapter 18

Remote Environments

NETWORKS ARE EFFECTIVE WHEN users are able to read their files and run their programs from remote locations. If you have users who often need remote access, you should consider configuring some Linux remote access services or even diskless workstations.

There are a number of different ways to access a Linux computer from a remote location. Several remote access services are controlled by the Extended Internet Services Daemon, `xinetd`. This daemon listens to ports such as those associated with the FTP and Telnet services. If you have the appropriate servers installed, `xinetd` starts these services upon request.

The `xinetd` daemon controls the operation of a number of remote access services, including Telnet, rsync, and POP3. Once installed, each of these services includes configuration files in the `/etc/xinetd.d` directory. You activate each service through these files; in many cases, you can also create a service-specific firewall.

Using the TCP Wrappers system, you can configure a detailed firewall for `xinetd` services. To regulate access to individual or all `xinetd` services, you customize `/etc/hosts.allow` and `/etc/hosts.deny`. You can still regulate access with an `iptables` firewall, as described in Chapter 17.

A number of `xinetd` services send messages in clear text. In Chapter 17, you've seen how this can put even your passwords at risk. One alternative for remote access to a Linux computer is the Secure Shell (SSH). The SSH daemon can be configured with private and public keys to encrypt messages over a network.

With all of these levels of security, it isn't always easy to diagnose a service problem. If users are having trouble accessing a server, you may need to check the available firewalls, one at a time. Other possibilities are that services are not active, or that various `iptables` commands or TCP Wrappers are blocking access.

One more remote environment is the diskless workstation. Once configured, you can set up as many terminals as you need. Each terminal gets access to an identically configured operating system. You can add read-write directories. This chapter covers the following topics:

- Using typical extended services
- Controlling access with TCP Wrappers
- Understanding the Secure Shell
- Troubleshooting access issues
- Configuring a diskless workstation

Using Typical Extended Services

Several basic services are controlled by `xinted`. These services include Telnet, POP3, and rsync, among others. For a list of currently installed `xinetd` services, review your `/etc/xinetd.d` directory.

The `xinetd` daemon includes two levels of configuration files. The first is `/etc/xinetd.conf`, which sets basic parameters. By default, it refers to configuration files in `/etc/xinetd.d` for service-specific parameters.

Many of the `xinetd` services are not encrypted. However, they do have their own levels of security. If you use the security measures associated with each service to limit their use to trusted users and computers, you limit the risks. As a Linux administrator, you need to make a judgment whether this is good enough for you and your organization.

NOTE *There are several* `xinetd` *services that were included in previous Red Hat distributions that are considered obsolete for Red Hat Enterprise Linux 3. Some are included in the Legacy Network Server package group. They include the remote shell (RSH) services, as well as Telnet. Red Hat Enterprise Linux includes a Kerberos version of Telnet.*

These services are different from those shown in the `/etc/rc.d/init.d` directory. Those services are independent and include their own connection and security mechanisms. In contrast, the services configured in the `/etc/xinetd.d` directory all use the `xinetd` service. It limits connections to keep your server from becoming overloaded.

The *xinetd* Configuration File

The first Extended Internet Services Daemon configuration file is `/etc/xinetd.conf`. The settings in this file set basic parameters for all services managed by `xinetd`. The default Red Hat Enterprise Linux configuration file is fairly straightforward, as shown in Figure 18.1.

FIGURE 18.1
/etc/xinetd.conf

```
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d

~
"/etc/xinetd.conf" 16L, 289C
```

Table 18.1 explains the parameters shown in this file. As you can see, this file uses `instances` to regulate the load on `xinetd`, specifies logging parameters, stops excessive connections, and includes the files in `/etc/xinetd.d`.

You can configure any of these parameters in other configuration files in the `/etc/xinetd.d` directory. When IP addresses are required, use regular or CIDR notation.

TABLE 18.1: *XINETD.CONF* **PARAMETERS**

COMMAND	DESCRIPTION
`instances`	Maximum number of active `xinetd` servers.
`log_type`	Specifies logging; `SYSLOG authpriv` specifies logging per `/etc/syslog.conf`, per Chapter 13.
`log_on_success`	Specifies logging information when a service starts and stops; useful parameters include `PID`, `HOST`, and `USERID`.
`log_on_failure`	Specifies logging information when a user requests a service that can't start; useful parameters include `HOST` and `USERID`.
`cps`	Regulates the rate of incoming connections; if connections exceed 25/sec, `xinetd` is disabled for 30 seconds, which can slow attempts to crack an `xinetd` service.
`includedir`	Every file in the specified directory is read as an `xinetd` configuration file.
`only_from`	Notes the IP addresses allowed to access the service.
`no_access`	Service is not allowed to computers with these IP addresses.
`access_times`	Specifies the times that access to the service is allowed; for example, `access_times = 08:00-23:00` means service is allowed between 8:00 a.m. and 11:00 p.m.

Activating *xinetd* Services

You activate an `xinetd` service in one of two ways: Either you directly edit the appropriate configuration file or you activate it with the appropriate `chkconfig` command. For example, if you've installed the `krb5-workstation-*` RPM package, you've installed the Kerberos version of the Telnet server. Open the `krb5-telnet` configuration file from the `/etc/xinetd.d` directory in a text editor. This and other `xinetd` configuration files contain a key parameter:

```
disable = yes
```

In other words, the service is disabled by default. You can enable it by changing this to

```
disable = no
```

You can make this change by editing this file directly in a text editor or by using the following command, where *`service_name`* is the name of the service (such as `rsync`) that you want to activate:

```
# chkconfig service_name on
```

Of course, you can reverse the process with the following command:

```
# chkconfig service_name off
```

After making a change, you may sometimes need to make `xinetd` reread the appropriate configuration file with the following command:

```
# service xinetd reload
```

Alternatively, you could reboot Linux, which would restart `xinetd` and make it reread the `/etc/xinetd.d` configuration files. But as you've probably noticed, rebooting Linux is rarely required.

TIP *The* `service` *command runs any of the scripts in the* `/etc/rc.d/init.d` *directory. For example, the* `service xinetd reload` *command is functionally equivalent to* `/etc/rc.d/init.d/xinetd reload`*.*

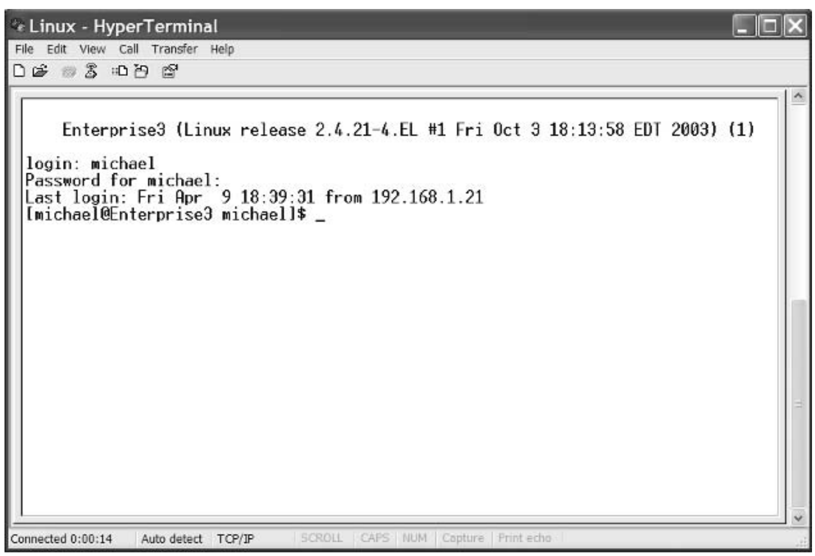
Kerberos Telnet

The Red Hat Enterprise Linux version of Telnet works in the same way as the legacy Linux Telnet service. It is still a simple way to connect to a remote computer. Many users are familiar with this service, and it is fairly easy to use. Telnet lets you quickly configure a number of different Linux terminals. In addition, you can practice configuring other `xinetd` services by using Telnet. While the legacy version sends messages, including passwords, in clear text; the Kerberos version encrypts messages using that protocol.

In Red Hat Enterprise Linux, the Telnet client RPM package is `telnet-*` ; the Telnet server package is part of the `krb5-workstation-*` RPM.

Once the network connection is made, Telnet is just like any other Linux command-line interface. One advantage is that Telnet is available on a variety of operating systems; Figure 18.2 shows an example of a Telnet connection to a Linux computer from a Windows XP operating system.

FIGURE 18.2 Telnet connection from a Microsoft computer



If you're having trouble with a Telnet connection or terminal, your Telnet client may be having a problem with the terminal messages sent from the Linux server that you're administering. One command that sets the environment variable to an older but standard terminal program is

```
TERM=vt100
```

As with other `xinetd` services, you need to activate it through the `/etc/xinetd.d/krb5-telnet` configuration file and reload `xinetd` before the Telnet server is active. Then you can access it from other computers with the `telnet` *`hostname`* command.

FTP Servers

The File Transfer Protocol, FTP, is one of the oldest protocols in the TCP/IP protocol suite. Because it is built for transferring files, it is still more efficient than newer protocols such as HTTP that can also transfer files. When I download the files to create Red Hat Enterprise Linux CDs, I use an FTP server.

We cover FTP servers in detail in Chapter 22, so we'll look at only the activation requirements here. The default Red Hat Enterprise Linux vsFTP server is not an `xinetd` service. Alternatively, one of the most common FTP servers is WU-FTP, maintained by Washington University in St. Louis. It's no longer included with Red Hat Enterprise Linux; but you can download it from the FTP site at `ftp.wu-ftpd.org` or the SpeakEasy RPM library at `www.rpmfind.net`. As you'd expect, when you install the `wu-ftpd-*` RPM, it installs a `wu-ftpd` configuration file in the `/etc/xinetd.d` directory.

As with other `xinetd` services, you need to activate it by setting `disable = yes` in the `wu-ftpd` configuration file.

Other Super Server Services

A number of other `xinetd` servers are available. They range from `finger`, which can give you more information about a specific user, to `pop3s` and `imaps`, which allow remote users to access their e-mail securely through your server. Some basic `xinetd` services are listed in Table 18.2. The list is not comprehensive.

TABLE 18.2: ASSORTED *XINETD* SERVICES

SERVICE	FUNCTION
`amanda`	Configures the advanced Maryland automatic network disk archiver for backups
`finger`	More information for a user, specified via `chfn`, stored in `/etc/passwd`
`imap`	Supports remote access to an IMAP4 mail server
`ipop3`	Supports remote access to a POP3 mail server
`rsync`	Allows automated use of the `rsync` service; see Chapter 14
`tftp`	Supports the Trivial File Transfer Protocol (TFTP) server; commonly used with diskless workstations
`krb5-telnet`	Sets up the Telnet server
`wu-ftpd`	Configures the WU-FTP server; see Chapter 22 (not available with Red Hat Enterprise Linux 3)

Controlling Access with TCP Wrappers

The best way to protect your system from crackers is to disable or uninstall as many services as possible. For example, a cracker can't use the `telnet` command to break into your computer if you don't have the Telnet server RPM installed.

If don't need an `xinetd` service immediately, one option is to deactivate it in the `/etc/xinetd.d` directory. In most cases, it's sufficient to leave all but required services disabled in the associated configuration files.

But some users need to get to their e-mail when they're in remote locations, and some need Telnet. For those who need access to larger files from remote locations, FTP servers are still the most effective way to transfer files over the Internet.

You can configure access control to all of these services using TCP Wrappers.

Regulating Access

You can minimize risks to your Linux computer in two different ways. First, you can regulate users and/or computers allowed to access a service through its configuration files, some of which are located in the `/etc/xinetd.d` directory. Other files are associated with different services and we address them in later chapters. Alternatively, you can use TCP Wrappers to regulate computer access through the `/etc/hosts.allow` and `/etc/hosts.deny` files.

You can add various rules to these files. These rules are read in the following order:

1. TCP Wrappers reads the `/etc/hosts.allow` file. If access is explicitly allowed, access is granted.
2. TCP Wrappers reads the `/etc/hosts.deny` file. If access is explicitly denied, users from the specified computer are not allowed to start the service.
3. If the computer or IP address is not found in either file, access is automatically granted, and `xinetd` starts the service.
4. If the computer or IP address is found in both files, the rule in `/etc/hosts.allow` comes first.

For example, if you configure rules that explicitly open a service to computer A in `/etc/hosts.allow` and then deny it in `/etc/hosts.deny`, computer A gets access.

In addition, any changes that you save to either file take effect immediately. You don't need to restart or reload the `xinetd` daemon.

The *xinetd* Firewall

A specific type of syntax is associated with `/etc/hosts.allow` and `/etc/hosts.deny`. First, as with other scripts, blank lines and comments that start with a `#` are not read. Each command line in these files should follow this configuration:

```
daemon: client: spawn command
```

In other words, when you specify a server daemon, you can associate it with a group of hostnames or IP addresses. When there is a match, you can also trigger a command, such as a message to the user or a log entry.

The simplest version of this command line is

```
ALL: ALL
```

which applies to all `xinetd` daemons and all computers. The options are complex, so it's easiest to examine the options one at a time.

TCP Wrapper Daemons

You can specify individual daemons, but keep in mind that the name of the daemon may not be what you expect. For example, the location of the Kerberos Telnet daemon is `telnetd`.

If you want to specify multiple daemons, just cite them together and separate the names of the daemons with a space. For example, the following line in `/etc/hosts.deny` blocks access to local Telnet and TFTP servers to all users:

```
telnetd in.tftpd: ALL
```

If in doubt on daemon names, refer to the configuration file in the `/etc/xinetd.d` directory. Each of these files includes the name of the daemon as the `server` variable.

TCP Wrapper Clients

You can specify the names of different computers, or `client` names, in TCP Wrappers commands by host or by IP address. Several wildcard parameters are available as well.

There are several ways to specify hostnames. You can specify them one at a time; for example, the following command prevents access to local TFTP and Telnet servers from the computers named sugaree and delilah:

```
telnetd in.tftpd: sugaree delilah
```

Or you can specify the fully qualified domain name (FQDN) of a computer, such as `sugaree.mommabears.com`. Wildcards are allowed with FQDNs; for example, just include the leading dot in `.mommabears.com` to apply the rule to all computers on the mommabears.com network.

It's possible to specify different computers; for example, the following line applies the rule to all computers on the `mommabears.com` network except `delilah.mommabears.com`:

```
in.tftpd: .mommabears.com EXCEPT delilah.mommabears.com
```

You can apply these principles to IP addresses; for example, the following line applies the rule to all computers on the 192.168.0.0 network except 192.168.0.102. Note the trailing dot in 192.168.0.; it applies to all computers with IP addresses between 192.168.0.0 and 192.168.0.255:

```
in.tftpd: 192.168.0. EXCEPT 192.168.0.102
```

Note *CIDR notation such as 192.168.0.0/24 does not work in the* `/etc/hosts.allow` *or* `/etc/hosts.deny` *files.*

Table 18.3 lists the wildcards that can be used in place of hostnames, FQDN, or IP addresses. They are fairly self-explanatory.

TABLE 18.3: TCP Wrapper Wildcards

Wildcard	Application
`ALL`	All computers, including the localhost.
`EXCEPT`	Exceptions to the rule.
`KNOWN`	Known computers—e.g., from DNS or `/etc/hosts`.
`LOCAL`	Computers with a single hostname; the name can't include a dot.
`PARANOID`	Computers where the hostname or FQDN does not match the IP address.
`UNKNOWN`	Computers not in the `/etc/hosts` or DNS databases.

TCP Wrappers Commands

Normally, all attempts to start an `xinetd` service are automatically added to `/var/log/messages`. You can use the `spawn` command to run another shell command as well. For example, use the following command to send an alert e-mail to the noted address:

```
telnetd: ALL: spawn /bin/mail -s "Telnet security alert" mj@example.com
```

Another common use is to append a special message to a log file in `/var/log` that identifies the date and time when someone tried to access the service.

Understanding the Secure Shell (SSH)

If you're concerned about someone intercepting your clear-text network communications, consider installing the Secure Shell (SSH). Because it encrypts your communications over any network, it's a viable alternative to the RSH commands, as well as Telnet.

SSH Installation

The SSH includes several component RPM packages, as shown in Table 18.4. Use the `rpm` commands discussed in Chapter 10 to install them as required. The basic packages should already be installed by default; the SSH `*askpass*` RPMs are part of the X Window package group.

TABLE 18.4: Secure Shell (SSH) Packages

Package	Function
`openssh-*`	Core files for SSH client and server
`openssh-askpass-gnome-*`	Files that support passphrase management inside GNOME
`openssh-askpass-*`	Files that support GUI management of SSH passphrases
`openssh-clients-*`	Client files for connecting to SSH servers
`openssh-server-*`	SSH servers

TIP *You can even use SSH on Microsoft Windows computers. As of this writing, a free version of the Open SSH package is available for download from Network Simplicity at* `www.networksimplicity.com`*. Once installed and configured, this client works just like the Linux version of SSH.*

SSH Configuration

The main SSH configuration file is `/etc/ssh/sshd_config`. While the default file works in most cases, you can adjust the settings in this file for special TCP/IP ports—for example, to limit access to different IP addresses, to adjust the size of encryption keys, to override RSH authentication, and to enable the use of Kerberos.

Once you have the appropriate packages installed, the next step is to create private and public encryption keys. You keep the private key secure on your Linux server. Public encryption keys allow others to scramble the messages they send to you. Alternatively, messages that you send are encrypted with the private key. They include the public key, which is used to unscramble the message only on the destination computer. These keys are based on random numbers so large (512 bits and more) that it would take weeks for a cracker with a personal computer to find.

Two basic SSH commands allow you to create private and public keys: `ssh-keygen -t rsa` and `ssh-keygen -t dsa`. These commands let you create keys based on the algorithm created by RSA Security or the Digital Secure Algorithm.

Both commands create the private and public keys, by default, in the `ssh` subdirectory of the user's home directory; thus the `~/.ssh file is created`, as listed in Table 18.5. When prompted, create a passphrase. If you don't set a passphrase, a cracker could steal your SSH private key. In some cases, this would allow the cracker to use your digital identity to use your credit cards or sign contracts in your name.

TABLE 18.5: DEFAULT SSH KEY FILES

ALGORITHM	PRIVATE	PUBLIC
DSA	`~/.ssh/id_dsa`	`~/.ssh/id_dsa.pub`
RSA	`~/.ssh/id_rsa`	`~/.ssh/id_rsa.pub`

Sample Session

Once you've installed the right RPMs on clients and servers and created the appropriate SSH keys, you're ready to begin using the Secure Shell. If desired, you can check to make sure the SSH server is running by issuing the `service sshd status` command.

Now you can connect directly to your account on another computer. For example, assume you are a user named cchavez and have an account on both computers. Run the `ssh sugaree.mommabears.com` command to connect to that computer. Be sure to substitute the computer name or IP address of your choice for `sugaree.mommabears.com`.

The first time you try to connect with `ssh` (or related commands), you'll see a message like the following:

```
The authenticity of host 'sugaree.mommabears.com (192.168.1.2)' can't be
 established.
```

```
RSA key fingerprint is34:21:d2:3c:34:83:40:23:d2:c2:9f:34:90:e3:a3.

 Are you sure you want to continue connecting (yes/no)?
```

Select Yes, and enter your password on the remote computer to complete the connection. You'll be able to work on the remote computer, and messages between your computers will be encrypted. Alternatively, you could log into a different account—say, vputin—as follows:

```
# ssh vputin@sugaree.mommabears.com
```

Alternatively, you could use the secure FTP service associated with SSH. If user vputin has a group of RPMs on his account and you have his password, you could use the secure FTP service to download files from his home directory on the remote computer. For example, the following commands log into that account and then download the source code for a new GNU C compiler to the local `/tmp` directory:

```
# sftp vputin@sugaree.mommabears.com
sftp> get gcc-3.9-8.src.rpm /tmp
```

Troubleshooting Access Issues

With all of these layers of protection, understanding an access problem can take some detective work. Here are some steps to follow if your users are having trouble accessing a service on your computer:

- Make sure the service is installed.
- Check to see that the service is active.
- Inspect security-related configuration files for the service.
- If it is an `xinetd` service, inspect the `/etc/hosts.allow` and `/etc/hosts.deny` files.
- Check the `iptables` firewall chains with the `iptables -L` command.

Check That the Service Is Installed

Checking for an installed service is fairly straightforward; as described in Chapter 10, you check the installation of an RPM package with the `rpm -q` *`packagename`* command.

Remember, it's common to organize services in separate client and server RPM packages. For example, there are separate client and server packages for Telnet, FTP, and SSH.

Verify That the Service Is Active

It's easy to use the scripts in the `/etc/rc.d/init.d` directory. As discussed in Chapter 13, every service daemon includes a script in this directory, which you can check with the `/etc/rc.d/init.d/`*`script`* `status` command. Alternatively, you could use the `service` *`script`* `status` command.

If you're wondering about an `xinetd` service, check the associated configuration file in the `/etc/xinetd.d` directory. By default, these services are set with `disable = yes`, which keeps a service closed.

And don't forget to use a tool such as `chkconfig` to make sure the service is active the next time you reboot Linux. For example, the following command verifies that `httpd` is active at runlevels 2, 3, and 5 when Linux starts:

```
# chkconfig --list 235 httpd on
```

The syntax for an `xinetd` service is slightly different, since these services are active at every runlevel where `xinetd` is active.

```
# chkconfig swat on
```

Inspect the Service-Specific Security Files

Many services include their own configuration files, which can limit or regulate access. Services such as Apache and Samba can be configured to limit access to different users and computers in their main configuration files. There are also `xinetd` services such as WU-FTP that have their own security files, such as `/etc/ftpaccess`. Service-specific security files are described in more detail in the chapters associated with each service.

Inspect the Extended *xinetd* Security Files

You've already learned how access can be limited through `/etc/hosts.allow` and `/etc/hosts.deny`. Just remember that similar commands can be used to limit access through the `/etc/xinetd.d` configuration files.

Check the Firewall *iptables* Chains

You can configure a firewall during or after the Red Hat Enterprise Linux installation process. After installation, you can use the `lokkit` or `redhat-config-securitylevel` utilities. Each of these Red Hat Enterprise Linux–specific tools offer default High and Medium security options, which lead to the same `iptables` chains.

NOTE *Of course, you can configure your firewall with your own* `iptables` *commands, using the techniques described in Chapter 17.*

The rules associated with both firewalls block access to your computer for most major TCP/IP ports. For example, to allow access to an Apache server on your computer, either set the appropriate `iptables` command, as described in Chapter 17, or use `lokkit` or `redhat-config-securitylevel` to customize the firewall to accept data to the appropriate TCP/IP port. (In this case, the right port is 80; you can look up different TCP/IP ports in `/etc/services`.)

Configuring a Diskless Workstation

A diskless workstation is also known as a *terminal*. It's a remote connection to an operating system. Linux happens to load that operating system in the memory of the local terminal. To set up diskless workstations, you need a server to share the operating system. It's also helpful to share directories for

user files. You also need terminals that can boot Linux directly from the network and get their IP address information from the server.

In principle, you can configure a server for diskless workstations on just about any current Linux system. However, you also need a terminal with a Pre-boot eXecution Environment (PXE) network card. With a PXE card, the BIOS can boot from the server, instead of from a local floppy or hard drive.

To make this work, you need to set up a dedicated directory on the server, the TFTP service for remote boots, NFS to share the operating system, and DHCP to assign IP addresses.

While Red Hat makes this process a bit easier with the Network Installation and Diskless Environment tool, the real driver behind Linux diskless workstations is the Linux Terminal Server Project, who are bringing Linux terminals on a large scale to schools, especially in the United States. When the Portland, Oregon, school district was faced with an audit for Microsoft Operating System licenses, it switched to Linux, using the economies of scale allowed by Linux diskless workstations.

NOTE *For more information on the Linux Terminal Server Project and their work with schools, navigate to* `www.ltsp.org` *and* `www.k12ltsp.org`.

Setting Up a Directory on the Server

You'll need to set up at least two directories on the server. One will contain the operating system that you want the terminal(s) to use. The other will contain any personal files for users of each terminal.

Before you start, have a model workstation in mind. It may be best to install Red Hat Enterprise Linux with just the packages that you need for this purpose. In that way, you'll need a minimum of space on the server, and your terminals can boot in a shorter period of time. You'll need to enable the SSH service on the model workstation. With the following steps, we'll show you how to set up the directory structure for a diskless workstation server:

1. Create an appropriate directory to store the operating system and user directory that you'll share, with a command such as:

```
# mkdir -p /terminal/RHEL3-WS
```

2. Set up a root subdirectory for the operating system, as well as a diskless directory for user files:

```
# mkdir -p /terminal/RHEL3-WS/root
# mkdir -p /terminal/RHEL3-WS/snapshot
```

3. Copy the files from the model workstation. The following command assumes that the name of the workstation is `model.example.com`:

```
# rsync -a -e ssh model.example.com:/ /terminal/RHEL3-WS/root
```

Remember, when you `rsync` all of the files from a workstation, the process can take some time. Once the copy process is complete, you can configure this server to share the files you need.

Starting TFTP for Access

In this case, the function of a TFTP (Trivial File Transfer Protocol) server is to support access to the shared operating system from the PXE boot environment. Once Linux has loaded on the terminal, NFS can be used to connect to the shared directories.

TFTP is an `xinetd` service. It is off by default. Therefore, you'll want to run the following command to make sure that it can run through `xinetd` now, and the next time you boot Linux:

```
# chkconfig --level tftp on
```

NOTE *We assume that you haven't changed the defaults for* `xinetd`*, which starts at runlevels 3, 4, and 5 by default.*

Configuring a DHCP Server for Diskless Access

A PXE boot system depends on a DHCP server for IP address information. That means you need to know how to configure a DHCP server, as described in Chapter 19. The key is that you need to configure DHCP addresses for each of terminals that you want to configure on your network.

In Chapter 4, we briefly described the process to configure PXE booting for network installations. The basic steps are the same for a diskless workstation server. You need to add the same basic commands to the main DHCP configuration file, `/etc/dhcpd.conf`.

```
allow booting;
allow bootp;
class "pxeclients" {
   match if substring(option vendor-class-identifier, 0, 9)="PXEClient";
   next-server 192.168.1.4;
   filename "linux-install/pxelinux.0";
}
```

The Red Hat system also requires that you configure static IP addresses for each diskless workstation. In the standard `dhcpd.conf` configuration file, there's already a standard set of commands for configuring an IP address for a DNS server. You can adapt these commands as needed for each diskless workstation.

```
host diskless1 {
     next-server server.example.com;
     hardware ethernet AB:CD:EF:12:34:56;
     fixed-address 192.168.1.122;
}
```

This particular command assumes a specific hardware address for the diskless workstation (`AB:CD:EF:12:34:56`). It assigns a specific hostname (diskless1) and IP address (192.168.1.122).

NOTE *You may be able to get the hardware address of the PXE network card from the BIOS. Alternatively, it may be listed when you start the PXE boot process.*

If you haven't already activated the DHCP server for your network, you'll want to do so as we describe in Chapter 19. The basic commands, as we describe in Chapter 13, are straightforward. These start the service and make sure it starts the next time you boot Linux:

```
# service dhcpd start
# chkconfig --level 35 dhcpd on
```

Configuring NFS on the Server

The most efficient file sharing system for Linux and Unix computers is the Network File System (NFS). We describe this service in detail in Chapter 22. Basically, you'll want to share the previously configured directories in `/etc/exports` and then restart the NFS service. You'll want to add the following lines to `/etc/exports`:

```
/terminal/RHEL3-WS/root      192.168.1.0(ro,sync,no_root_squash)
/terminal/RHEL3-WS/snapshot  192.168.1.0(rw,sync,no_root_squash)
```

Be careful; extra spaces in the wrong places in this file can lead to errors. For a detailed discussion of these commands, see our description of NFS in Chapter 22.

Setting Up the Network Booting Service

Now that you've configured your directories, copied a model version of the Linux operating system, configured TFTP, DHCP, and NFS, you're ready to put it all together. This is where the Red Hat Network Installation and Diskless Environment tool can help. This tool is also known by its basic command, `redhat-config-netboot`, as shown in Figure 18.3. You can also start it from a GNOME GUI with the Main Menu ➤ System Settings ➤ Server Settings ➤ Network Booting Service command. First, you'll set up the server, and then you'll add as many diskless workstations as you need.

NOTE *If this is the first time you've opened* `redhat-config-netboot`*, you'll see the First Time Druid. Click Cancel to open the window shown in Figure 18.3.*

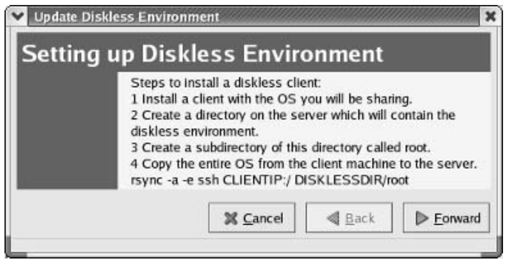
FIGURE 18.3
Configuring a diskless environment

Configuring the Diskless Environment

In this section, we'll configure the diskless environment with the Red Hat `redhat-config-netboot` tool. Start from the Network Installation and Diskless Environment window shown in Figure 18.3, and follow these steps.

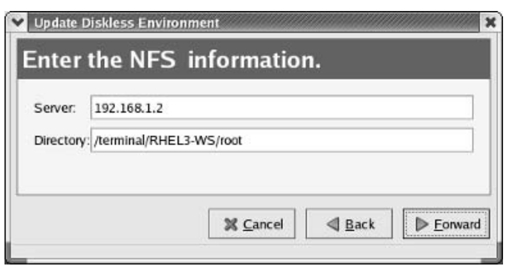
1. Click Configure ➢ Diskless. This opens the Configure Diskless Environment window.
2. Click Add. This opens the window shown in Figure 18.4, which lists some of the steps that you've taken so far.

Figure 18.4
Setting Up Diskless Environment



3. Click Forward. This opens the Diskless Identifier window. The Name you type in will be added as a subdirectory to `/tftpboot/linux-install`. When a diskless workstation first boots, it will look to this directory for basic start files. Enter the Description of your choice.
4. Click Forward. This opens the Enter The NFS Information window shown in Figure 18.5. Enter the hostname or IP address for your NFS server, as well as the root directory you configured earlier, in this case, `/terminal/RHEL3-WS/root`.

Figure 18.5
Setting Up Diskless Environment



5. Click Forward. If you've properly shared the directory over NFS, you'll now be able to select the kernel of your choice, as shown in Figure 18.6. If you've selected a simple model workstation, you may have only one kernel to choose from.

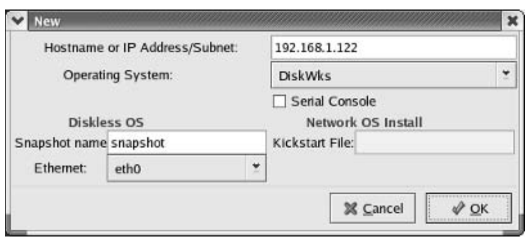
Figure 18.6
Selecting a Kernel

Once you click Apply, the `redhat-config-netboot` tool copies any files that your diskless workstation may need to the `/tftpboot/linux-install` directory. Naturally, you can do this from the command line interface, but you'll have to remember each file and each directory.

Adding a Diskless Workstation Host

Finally, you can add the diskless workstations to your server. Return to the Network Installation and Diskless Environment window shown in Figure 18.3. Click New; this opens the New window shown in Figure 18.7. In this window, you can enter the parameters for your new diskless workstation. Those shown in the figure are based on the parameters we set earlier.

Figure 18.7
Configuring a diskless host



For each diskless workstation, you'll want to:

- Specify the same IP address as that configured in `/etc/dhcpd.conf`.
- Note the same subdirectory name as shown in the `/tftpboot/linux-install` directory.
- Set the same Snapshot name as the subdirectory specified in `/terminal/RHEL3-WS`.

Booting a Diskless Workstation

Finally, we're ready to boot a diskless workstation. Modern terminals allow you to boot from a PXE network card through the appropriate BIOS. While BIOS menus vary by computer, you can usually set your system to boot from the network card using the same menu as you may have used to boot from the first installation CD.

When you boot through the PXE card, you should see a series of messages that include the hardware address of that card. When it detects and receives a message from your DHCP server, you may

see the IP address that you configured flash briefly. Then the diskless workstation boots from the `/tftpboot/linux-install` directory. You can then load the operating system using appropriate NFS commands.

Summary

Users often need to get their files from remote locations. The files an engineer has on his laptop may not be the configuration files he needs to solve a client's problem. Linux provides a selection of different remote access services. Many of them are part of the Extended Internet Services Daemon, `xinetd`.

The `xinetd` daemon controls access to and starts various services on demand. Access is controlled through `/etc/xinetd.conf` and individual service files in the `/etc/xinetd.d` directory. New `xinetd` services are disabled by default. Three major `xinetd` remote access services are WU-FTP, Telnet, and POP3.

Access to `xinetd` services is controlled through TCP Wrappers, which depends on configuration commands in `/etc/hosts.allow` and `/etc/hosts.deny`. You can configure commands for specific services, addressing specific computers or networks. When there is a match, you can also set these commands to run shell commands that might send you a warning or send the information to a log file.

One alternative service that encrypts remote communication is the Secure Shell (SSH). The various `openssh-*` RPM packages allow you to use RSA or DSA encryption for network communication. With this type of public/private key system, it is important for you to protect your private key with a passphrase. You can use SSH commands to open your account on remote computers, or even connect securely to a SSH-enabled FTP server.

Troubleshooting remote access issues can be problematic, because there is a wide range of available firewalls. A service might not be installed or active. Many services have their own security-related configuration files, and you'll need to check those files. You can protect `xinetd` services through `/etc/hosts.allow` and `/etc/hosts.deny`. And, of course, you can configure firewalls with `iptables`.

Perhaps the ultimate in remote services is the diskless workstation. With an NFS server, you can set up a series of diskless workstations. The operating system is shared as a read-only directory. You can give each workstation its own read-write directory for user data.

In Chapter 19, we'll look at detailed configuration requirements for two major Linux servers and their clients: the Domain Name Service (DNS) and the Dynamic Host Configuration Protocol (DHCP).

Chapter 19

DNS and DHCP

Two key services can help every Linux computer manage hostnames and IP addresses. The Domain Name Service (DNS) allows you to configure a database of hostnames or domain names and IP addresses. The Dynamic Host Configuration Protocol (DHCP) enables you to ration IP addresses by leasing them to different computers on your LAN. As with most other Linux services, both DNS and DHCP include a client and a server.

The Linux DNS server is based on Berkeley Internet Name Domain (BIND) software and can be configured through a series of files in `/etc` and `/var/named`. It is also known as a *nameserver*, using the `named` daemon. Any Linux computer that is configured to use TCP/IP is by default configured as a DNS client. When you look for a website, your computer acts as a DNS client. It looks to the DNS server for the associated IP address so it knows where to send its message on the Internet. While Red Hat includes a GUI tool to configure DNS servers, the version available as of this writing for Red Hat Enterprise Linux 3 is not reliable. I therefore recommend that you do not use this tool, and I do not describe how to use it in this book.

A DHCP server can lease IP addresses and provide other key information that allows your computer to define itself on your LAN. DHCP servers can be configured with information that enables your computer to access external networks, find other important servers, and more. Red Hat Enterprise Linux has changed the name of its DHCP client a number of times in recent years, but the functionality is still the same. It gets IP addresses from a DHCP server, and it collects any other information available from that server. This chapter covers the following topics:

- Configuring a DNS server
- Setting up a DHCP server
- Using a DNS client
- Working with DHCP and BOOTP clients

Configuring a DNS Server

A Domain Name Service (DNS) server is a flexible database of fully qualified domain names (FQDN), such as `www.sybex.com`, and IP addresses, such as 63.99.198.12. The Linux version of DNS is the `named` daemon, which is based on BIND, which powers most of the DNS servers on the Internet.

No one DNS server can hold all the FQDNs and IPv4 addresses on the Internet. If a DNS server does not have a FQDN in its database, it can refer to other DNS servers. Once the server finds the right IP address, it adds the FQDN and IP address to its database.

DNS is configured through the basic configuration files `/etc/named.conf` and `/etc/named.custom`, as well as through detailed configuration files in the `/var/named` directory. It is still best to edit these files directly to configure DNS.

Packages

Not all of the RPM packages that you need for DNS are installed by default. The required packages are listed in Table 19.1; as you may remember from Chapter 10, you can use the `rpm -q` *`packagename`* command to see if they're installed. Once the packages are installed, you can use the `rpm -ql` *`packagename`* command to see the associated files.

TABLE 19.1: DNS RPM PACKAGES

PACKAGE	FUNCTION
`bind-*`	The DNS name server software
`bind-utils-*`	DNS tools such as `dig` and `host`; installed by default
`caching-nameserver-*`	Basic configuration files for a caching DNS server; includes sample `/etc/named.conf` and `/var/named/localhost.zone` files
`redhat-config-bind-*`	The Red Hat GUI DNS configuration tool

The Red Hat GUI DNS configuration tool is fairly new. It has problems. It cannot configure all the settings associated with a standard LAN. If you don't want to use it to configure your DNS server, you should uninstall it with the `rpm -e redhat-config-bind-*` command. Otherwise, another administrator in your organization may accidentally overwrite any changes you make to the DNS configuration files.

TIP *There are a number of problems associated with the version of* `redhat-config-bind` *associated with the original release of Red Hat Enterprise Linux 3 (*`2.0.0-14`*). An upgrade was not available through the Red Hat Network as of this writing. Until an upgrade is available, I recommend you configure your DNS server directly and uninstall* `redhat-config-bind`*.*

DNS Concepts

As we mentioned earlier, no single DNS server can contain the database of FQDN and IP addresses for the entire Internet. Because of the volume of associated data, it isn't practical to centralize DNS information. Therefore, DNS servers are organized in *zones*. Each DNS server has its *zone of responsibility*. DNS zones are based on the way FQDNs are organized.

Start with a basic FQDN, `www.mommabears.com.` This includes a root zone, which is not the `.com` but the period to the right of the `.com`.

NOTE *The root DNS servers are listed in* `/var/named/named.ca`*, which is part of the* `caching-nameserver-*` *RPM package.*

The next phrase may be `.com`, `.net`, `.org`, and so on; these are known as *top-level domains*. In this case, mommabears is a *subdomain* of `.com`, and `www` is the name (or more likely the alias) of a computer with the Momma Bears' web server.

A master DNS server on the `mommabears.com.` network would be the authoritative server for that zone. Conversely, `mommabears.com.` is the Forward (or Primary) Master Zone for that DNS server.

These database zones aren't complete unless you can reverse the process. In other words, you should be able to find an IP address from a FQDN—and you should be able to reverse the process by finding a FQDN from an IP address. The reverse database is known as a *Reverse Master Zone*.

You can configure four different types of DNS servers. As you'll recall from Chapter 16, the IP address of any DNS server that you use should be listed in `/etc/resolv.conf`.

Master A master DNS server is the authoritative server for a specific zone, such as `sybex.com`. Queries for IP addresses from computers on the `sybex.com` network normally go to this server. Other DNS servers refer to this master for addresses of other networks and computers within `sybex.com`.

Slave Queries for IP addresses from within `sybex.com` can go to this server; it gets its FQDN/IP address database from a master DNS server.

Caching-only A caching-only DNS server stores recent requests for IP addresses. If you have a caching-only DNS server on your LAN and your DNS server is on a remote network, your computers can often get quicker answers by using the caching-only DNS server. The default `/etc/named.conf` file is designed for a caching-only nameserver that's connected to the Internet.

Forwarding A forwarding DNS server does not store any FQDN/IP address information. It does store the IP addresses of other DNS servers in `/etc/named.conf`.

Initial DNS Configuration

I encourage users to configure Linux services at the command-line interface. If you do so, you learn more about the service and can better customize the service for the network. In the following sections, we'll show you how to edit the standard DNS configuration files to create each of the four types of DNS servers that we just described.

TIP *If you're going to use* `redhat-config-bind` *and work at the command line, don't edit* `/etc/named.conf` *directly. If you use Red Hat's GUI tool, be aware that it writes its changes to this file. Add your text configuration changes to* `/etc/named.custom`*.*

DNS Configuration Files

Several configuration files are required for a DNS server: `/etc/named.conf`, `/etc/named.custom`, and database files in the `/var/named` directory. It is best to edit these files directly with the text editor of your choice. It's helpful to examine each of these files in detail. But first, we list the basic DNS configuration files in Table 19.2.

In this section, we'll look at configuring `/etc/named.conf` for a standard DNS server. Later, we'll show you how to configure the database files in `/var/named` for a standard DNS server.

Table 19.2: DNS Database Files in */etc*

File	Function
`sysconfig/named`	If you want to set up DNS configuration files in nonstandard locations, document it here.
`named.conf`	The basic DNS configuration file; you can edit it directly or through `redhat-config-bind`.
`named.custom`	If you use `redhat-config-bind`, you can use this file to add more DNS settings.
`rndc.key`	The authentication key that supports DNS requests; configured through `/etc/rndc.conf`.

As our intent is to describe the standard DNS configuration, we will leave the `/etc/sysconfig/named` file blank. The best use of this file is to help secure your system by configuring DNS in a *chroot jail*. Briefly, with this configuration, anyone who breaks into the DNS directory tree is kept away from any other critical files on that computer. For more information on this process, refer to the Chroot-BIND HOWTO at `www.tldp.org`. First we'll look at configuring `/etc/named.conf` for a regular DNS server and then create the required data files in the `/var/named` directory. Then we'll put it all together, describing what you need to do to configure the four basic types of DNS servers.

Configuring a DNS Server in */etc/named.conf*

The main DNS configuration file is `/etc/named.conf`. The default version of this file is shown in Figure 19.1. You can just as easily create this file in any text editor. However, if you use `redhat-config-bind`, be sure to add any additional parameters to `/etc/named.custom`.

Figure 19.1
/etc/named.conf

```
// generated by named-bootconf.pl
options {
        directory "/var/named";
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query-source address * port 53;
};
// a caching only nameserver config
controls {
        inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
        type hint;
        file "named.ca";
};
zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
        allow-update { none; };
};
include "/etc/rndc.key";
```

It's useful to break down this file, command by command. The order of commands may vary, depending on the installed version of the `caching-nameserver` RPM. The version we're analyzing is 7.2–7. If your version is different, you may find commands in different locations in the file.

The first command in the file is easy to miss, interspersed with the initial comments. This particular `options` command sets `/var/named` as the data directory for this DNS server:

```
options {
     directory "/var/named";
```

If you're at all familiar with scripts or programming, you'll notice this isn't the entire command. As noted in the comment, if you have a hardware firewall between this computer and other DNS servers, you'll want to activate the following command by deleting the two forward slashes:

```
// query-source address * port 53;
```

Next, the following command limits access to the `rndc` command to users on the local computer. However, it does not limit users to access the local computer remotely using a service such as SSH.

```
controls {
     inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
```

NOTE *Many administrators use SSH (see Chapter 18) to connect to remote DNS servers. However, you could also set the* `controls` *line in* `/etc/named.conf` *to the IP address and name of another computer on your LAN. You could then use* `rndc`*, the remote name daemon control utility, to manage your DNS server remotely.*

Now the following commands allows your DNS server access to the DNS database servers for the Internet. These servers are listed in the `/var/named/named.ca` file from the `caching-nameserver-*` RPM.

```
zone "." {
     type hint;
     file "named.ca"
};
```

NOTE *If you used* `redhat-config-bind`*, you should find this* `zone` *command in your* `/etc/named.custom` *file.*

This DNS server has basic zones of authority. The first two stanzas are related to the localhost computer, IP address 127.0.0.1. This is by default a zone of authority. The domain of your LAN—in this case, `example.com`—is a second zone of authority. Inverse zones, as indicated by the `in-addr.arpa` statement, are also an important part of the DNS database. Because these are reverse IP addresses, the 1.168.192.in-addr.arpa zone is based on the 192.168.1.0 network address. However, you can assign the name of your choice; in this case, it's `example.com.rr.zone`.

```
zone "localhost" IN {
     allow-update { none; };
     type master;
```

```
        file "localhost.zone"
};
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "0.0.127.in-addr.arpa.zone"
};
zone "example.com" IN {
        type master;
        file "example.com.zone"
};
zone "1.168.192.in-addr.arpa" IN {
        type master;
        file "exmple.com.rr.zone"
};
```

Finally, you may need to add the following `include` directives. The first directive adds the contents of the `/etc/named.custom` configuration file. As configured at the end of the file, the Red Hat GUI DNS configuration tool does not overwrite these commands.

```
include "/etc/named.custom";
include "/etc/rndc.key";
```

The *rndc* Control Key

The Remote Name Daemon Control Utility is known as `rndc`. The Red Hat Enterprise Linux standard DNS packages contain files that refer to a standard `rndc` encryption key, stored in `/etc/rndc.key`. This encrypts communication to and from a DNS server. As long as the `include "/etc/rndc.key";` command is present in both the `/etc/named.conf` and `/etc/rndc.conf` files, you'll be able to start and communicate with your DNS server.

As this is a standard encryption key, anyone with a copy of Red Hat Enterprise Linux 3 will have access to this key. If you feel the need to secure your DNS server, you'll want to change this key. The following command automatically (`-a`) sets up a new key in `/etc/rndc.key`, with a key size of 512 (`-b`) bits:

```
# rndc-confgen -a -b 512
```

Note *This command leads to a minor error in the* `/etc/rndc.key` *file. You'll need to make sure this line starts with* `key "rndckey"` *and not* `key "rndc-key"`.

DNS Database Files

The database files that support a DNS server are by default located in `/var/named`. The names of these files depend on the name of your domain, the IP address of your network, and whether you're supporting a regular or a caching-only DNS server. Some of the files you may see are listed in Table 19.3.

Table 19.3: DNS Database Files in */var/named*

File	Function
netaddr.in-addr.apra.zone	Specifies the reverse zone file for the LAN, where *netaddr* is the first three octets of a network address, backward; for example, for the 192.168.4.0 network, this file would be 4.168.192.in-addr.arpa.zone.
domain.zone	Specifies the zone file for the LAN, where an address such as example.com is substituted for *domain*.
localhost.zone	Specifies the zone file for localhost.
named.ca	Lists Internet root servers; from the caching-nameserver-* RPM.
named.local	Specifies the PTR, a reverse zone record for localhost.

In this section, we examine the default localhost files, so you can use them as a guide. Then we show you how to create DNS database files for an example.com network. For this purpose, we'll create the example.com.zone and example.com.rr.zone files in the /var/named directory.

localhost Zone Files

Now let us examine a forward and a reverse zone database file. Start with Figure 19.2, which illustrates the zone file for the default localhost.

Figure 19.2
/var/named/localhost.zone

```
$TTL    86400
$ORIGIN localhost.
@                       1D IN SOA       @ root (
                                        42              ; serial (d. adams)
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                        1D IN NS        @
                        1D IN A         127.0.0.1
~
~
~
~
```

As you can see, this file contains a number of strange-looking commands. We've listed these commands in Table 19.4.

Table 19.4: *localhost.zone* Commands

Command	Function
$TTL	Specifies the length of time for which the data in this file is good. Default is seconds; 86400 is three days. May also be shown as $TTL 3D.
$ORIGIN	Allows you to list hostnames that may not correspond to the FQDN for that computer. Otherwise, your DNS server would assume this computer's FQDN is localhost.mommabears.com.

Continued on next page

Table 19.4: *localhost.zone* **Commands** *(continued)*

Command	Function
`@`	Adds the `$ORIGIN` command or the domain specified in `/etc/named.conf`.
`IN`	Notes a standard data record; also known as INternet class data.
`SOA`	Start of Authority; specifies key information about this database.
`@ root`	Messages are sent to the root user on the localhost computer.
Serial Number	The 42 in `localhost.zone` is the serial number associated with this file. Normally this includes the date and revision number; the first revision of the file created on August 9, 2004, is 200408091.
`Refresh`	Notes the time between checks to the primary DNS server for this zone, in seconds.
`Retry`	Sets a time to try to contact a DNS server if the first attempt fails. If a refresh attempt can't reach a DNS server, try again after this many seconds.
`Expiry`	Notes a stop time; If refresh and retry attempts don't reach a DNS server, stop activity on this zone after this much additional time; in seconds.
`Minimum`	Sets a minimum TTL for the data in this file.
`NS`	Specifies the nameserver for this data, in this case, also localhost.
`A`	Address record; specifies the IP address associated with this name.

The reverse database file for localhost is `named.local`. The commands in this file are straightforward; there's one new command relative to `localhost.zone`.

```
1    IN     PTR     localhost.
```

Domain Zone Files

If you want to practice configuring a DNS server, you can get your own domain name, or you can use the `example.com` domain. This domain has been explicitly reserved for experiments just like this. In this section, we create `example.com.zone` and `example.com.rr.zone` files for this domain. I've set up my computers on the `example.com` network, as shown in Figure 19.3.

Tip *The syntax associated with this file requires precision. For example, if you forget a dot at the end of any FQDN, you'll get an error message such as* `Host example.com not found`.

We'll analyze only those variables that are different from the previously analyzed `localhost.zone` file. This first line sets the time to live. The second line describes the zone, as governed by the `enterprise3d.example.com` computer. The administrative e-mail for this server is `dnsadmin@example.com`. The `dns.example.com` statement is the syntax that specifies the noted email address.

```
@     IN     SOA     enterprise3d.example.com. dnsadmin.example.com.  (
```

FIGURE 19.3
/var/named/example.com.zone

```
;
;       A test zone file, using the example.com domain
;
;       This is the database file for the computers on Mike's network
;               on the example.com zone
;               (Substitute your domain name)
;
$TTL 3D
@     IN     SOA     enterprise3d.example.com. dnsadmin.example.com.  (
                     200404131  ; serial number = today's date + rev number
                     24H        ; refresh frequency (24 hours)
                     4H         ; retry frequency (4 hours)
                     3W         ; data expiration period (3 weeks)
                     3D )       ; TTL of at least 3 days

      IN     NS      enterprise3d.example.com.  ; The FQDN of the nameserver

      IN     MX      10 mail.example.com.       ; Higher priority mailserver
      IN     MX      20 mail2.example.com.      ; Backup mailserver

allaccess    IN      A     192.168.1.53
enterprise3d IN      A     192.168.1.4
enterprise3  IN      A     192.168.1.13
bluesman     IN      A     192.168.1.21

mail         IN      CNAME enterprise3d
mail2        IN      CNAME enterprise3
ftp          IN      CNAME enterprise3d
www          IN      CNAME enterprise3d
~
```

The nameserver for this network, as described by NS, is `enterprise3s.example.com`. The following lines set up two different mail servers; the lowest number gets first priority:

```
MX     10 mail.example.com ; Higher priority mailserver
MX     20 mail2.example.com ; Backup mailserver
```

Next, I've set up the computers on my network. The format is straightforward; for example, the following statement assigns a specific IP address record (`A`) to the allaccess computer. As this file configures the `example.com` network, it's assumed that this address record is for the `allaccess.example.com` computer.

```
allaccess     IN     A     192.168.1.53
```

If you configure different servers on the same computer, you should set up aliases. For example, the following commands set up the previously noted mail servers on different computers on this network:

```
mail     IN     CNAME    enterprise3d
mail2    IN     CNAME    enterprise3
```

Now look at the reverse database file in Figure 19.4. As described earlier, we've configured this in the `example.com.rr.zone` file. As you can see, it includes the same basic commands as in a regular DNS database file. The `PTR` records may appear a bit strange.

To find the IP address, you need the `PTR` record number as well as the name of the file. For example, the first PTR record line in Figure 19.4 starts with 4, and the FQDN is `enterprise3d.example.com`. When correlated with the name of the file, `example.com.rr.zone`, your DNS server can identify the IP address of `enterprise3d.example.com`: 192.168.1.4.

FIGURE 19.4
A reverse zone file

```
;
;       A test reverse zone file, using the example.com domain
;
;       This is the database file for the computers on Mike's network
;               on the example.com zone
;               (Substitute your domain name)
;
$TTL 3D
@       IN      SOA     enterprise3d.example.com. root.example.com.  (
                        200404131  ; serial number = today's date + rev number
                        24H        ; refresh frequency (24 hours)
                        4H         ; retry frequency (4 hours)
                        3W         ; data expiration period (3 weeks)
                        3D )       ; TTL of at least 3 days

        IN      NS      enterprise3d.example.com.  ; The FQDN of the nameserver

4       IN      PTR     enterprise3d.example.com
13      IN      PTR     enterprise3.example.com
21      IN      PTR     bluesman.example.com
53      IN      PTR     allaccess.example.com
~
~
"example.com.rr.zone" 21L, 756C
```

Starting and Testing Your DNS Server

Once you've configured DNS, you'll want to try your new server. The easiest way to do this in Red Hat Enterprise Linux is with the DNS service script. Remember, `named` is the daemon that runs the Linux DNS server.

```
# service named start
```

NOTE *One common problem is an* `rndc: connect failed: connection refused` *message. There are several possible causes. Your hostname may be missing from* `/etc/hosts`*. You may not have an* `include "/etc/rndc.key";` *statement in your* `/etc/named.conf` *or* `/etc/rndc.conf` *files, which makes sure you have the same name server control key in each file.*

If you've already started your DNS service, you can still edit the `/etc/named.*` configuration files and the `/var/named` database files. You don't need to restart the `named` daemon. You can reload the data with the following command:

```
# rndc reload
```

You should check to see what your DNS server has done with the data. You can do this with the `host -l` command. As we've configured the `example.com` domain, the command is `host -l example.com`. We've illustrated the result in Figure 19.5. It should look familiar; you should be able to correlate the output with the information in the `/var/named/example.com.zone` database file.

Next, you'll want to see if it works. As described in Chapter 16, the IP addresses of DNS servers are normally listed in `/etc/resolv.conf`. Once you've started your `nameserver`, you can see how it works. Try the `dig` command, the DNS lookup utility, to look up a specific FQDN on your network or the Internet. Figure 19.6 shows how this works. Note the `SERVER` line near the bottom of the figure, which illustrates that this comes from a DNS server on a computer on my local network, with an IP address of 192.168.1.4.

FIGURE 19.5
Checking your DNS server database

```
[root@Enterprise3 root]# host -l example.com
example.com SOA enterprise3d.example.com. root.example.com. 200404131 86400 1440
0 1814400 259200
example.com name server enterprise3d.example.com.
example.com mail is handled by 10 mail.example.com.
example.com mail is handled by 20 mail2.example.com.
allaccess.example.com has address 192.168.1.53
bluesman.example.com has address 192.168.1.21
enterprise3.example.com has address 192.168.1.13
enterprise3d.example.com has address 192.168.1.4
ftp.example.com is an alias for enterprise3d.example.com.
mail.example.com is an alias for enterprise3d.example.com.
mail2.example.com is an alias for enterprise3.example.com.
www.example.com is an alias for enterprise3d.example.com.
example.com SOA enterprise3d.example.com. root.example.com. 200404131 86400 1440
0 1814400 259200
[root@Enterprise3 root]#
```

FIGURE 19.6
The *dig* command

```
[root@Enterprise3 root]# dig allaccess.example.com

; <<>> DiG 9.2.2 <<>> allaccess.example.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19938
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;allaccess.example.com.         IN      A

;; ANSWER SECTION:
allaccess.example.com.  259200  IN      A       192.168.1.53

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      enterprise3d.example.com.

;; ADDITIONAL SECTION:
enterprise3d.example.com. 259200 IN     A       192.168.1.4

;; Query time: 79 msec
;; SERVER: 192.168.1.4#53(192.168.1.4)
;; WHEN: Wed Apr 14 19:04:12 2004
;; MSG SIZE  rcvd: 98

[root@Enterprise3 root]#
```

If you haven't configured the `example.com` domain for your LAN, try the `dig example.com` command. You'll see that its official DNS servers are associated with the IANA, the Internet Assigned Numbers Authority.

If you're satisfied with the result, remember to make sure `named` starts the next time you restart Linux. Use the `chkconfig --level 235 named` command to ensure this daemon starts at runlevels 2, 3, and 5.

TIP *When you start DNS during the boot process, you can also check startup messages in the* `/var/log/messages` *file. If there is a problem, such as a syntax error in* `/etc/named.custom`*, you'll see an indication here.*

A DNS Forwarding Server

If you're configuring a network on a larger enterprise, someone may have already configured DNS servers for your organization. If it's outside your LAN, you can configure a forwarding-only DNS server, in `/etc/named.conf`.

TIP *If you've already configured a regular DNS server and want to try other types of DNS servers, back up your* `/etc/named.conf` *configuration and* `/var/named` *database files.*

Alternatively, if this is a forwarding-only DNS server, for all domains you may see a different options statement, where the forwarders are the IP addresses of other DNS servers, or perhaps for your ISP:

```
options {
        directory "/var/named";
        forward only;
        forwarders {
                10.11.12.13;
                10.11.12.14;
                10.11.12.15;
        };
```

You can test the result with the `host` and `dig` commands, as described in the previous section. If the `host -l example.com` command doesn't work, the DNS forwarding servers you've added don't contain information about your `example.com` domain.

A DNS Caching-Only Nameserver

Most users connect to the Internet using an ISP. These providers have DNS servers that you can use to help navigate the Internet. However, you don't absolutely need the DNS servers associated with an ISP. You can configure your system as a caching-only DNS nameserver.

If you're configuring a caching-only DNS, the default `/etc/named.conf` configuration file should work. If needed, you can set up a new copy of this file, using the following steps:

1. Back up your current `/etc/named.conf` configuration file.
2. Connect to the source of your installation files. For the purpose of this exercise, assume that it's mounted on the `/mnt/inst` directory.
3. Uninstall the base RPM package with the `/etc/named.conf` configuration file with the following command:

   ```
   # rpm -e caching-nameserver
   ```

4. Install the original `caching-nameserver` RPM with the following command:

   ```
   # rpm -Uvh /mnt/inst/RedHat/RPMS/caching-nameserver-*
   ```

   Now you should have an original copy of the default `/etc/named.conf` configuration file. You should now be able to activate a caching DNS server on your computer. The only question is whether there may be a firewall that blocks your requests to other DNS servers. If so, remove the two forward slashes from in front of this command, which serve as comment characters.

   ```
   // query-source address * port 53
   ```

As you saw in Chapter 17, even the default high-security Linux `iptables` firewall allows DNS requests through TCP/IP port 53. Without this command, requests to other DNS servers may be blocked since the latest versions of BIND often use other ports. When you're ready, you can complete the process with the next steps.

5. Include the IP address of your computer as the DNS server in `/etc/resolv.conf`.
6. Start the `named` daemon.

The key with all of this is the `zone "." IN` stanza. With the root nameservers in the `/var/named/named.ca` file, your caching DNS server can now use the root servers on the Internet to find the IP addresses you need.

You can test the result with the `host` and `dig` commands, as described in the previous section. If the `host -l example.com` command doesn't work, the DNS forwarding servers you've added don't contain information about your `example.com` domain.

A DNS Slave Server

To configure a secondary (slave) DNS server, you'll want to take the same basic steps as with a regular DNS server. The difference is that you'll want to identify your zone as `type slave`. If you want to configure the `example.com` network, you'd also need to use the `masters` command to call the master DNS server.

```
zone "example.com" IN {
     type slave;
     file "example.com.zone";
     masters { 192.168.1.4 };
};
```

This assumes that you already have a master DNS server on IP address 192.168.1.4. Once you've activated the named daemon on the local computer, you can test the result. Run the `host` and `dig` commands, as described in the previous section. For example, if the `host -l example.com` command doesn't work, the DNS forwarding servers you've added don't contain information about your `example.com` domain.

Using a DNS Client

If you've configured your computer to communicate on the Internet, you've already set it up as a DNS client. When you try to access another computer, you look for a database of hostnames and IP addresses.

As discussed in Chapter 16, there are two databases of hostnames and IP addresses: `/etc/hosts` and the DNS servers listed in `/etc/resolv.conf`. If you've configured a DNS server on your own network, make sure its IP address is included in `/etc/resolv.conf`. Otherwise, add the DNS servers assigned by your ISP to this file.

The databases used by your computer are determined by your `/etc/host.conf` file. This file normally includes one line: `order hosts,bind`. This means your computer first looks for IP addresses in

`/etc/hosts` before searching any DNS servers listed in `/etc/resolv.conf`, which you can configure per the instructions in Chapter 16.

No additional configuration is required.

Setting Up a DHCP Server

The Dynamic Host Configuration Protocol (DHCP) can automatically give all TCP/IP computers on your network the information they need to communicate. This includes the routers, the DNS servers, other name type servers, as well as basic IP addressing information.

To set up a computer as a DHCP server, you'll need to make sure the network card can handle multicast requests. If you have older Microsoft Windows computers, you should also set up the broadcast address as a dedicated route. Then you can configure the DHCP configuration file, `/etc/dhcpd.conf`. If you want to use your DHCP server for remote networks, you'll also have to configure `dhcrelay` on the router/gateway between your LANs. The `dhcrelay` daemon supports the BOOTP protocol.

The DHCP server is installed as a default part of the Network Server package group. Alternatively, you can install it using the appropriate `rpm` command. As of this writing, there is no Red Hat GUI tool you can use to configure a DHCP server.

Basic Configuration

Before you start configuring `/etc/dhcpd.conf`, you need to check a couple of things about your network configuration. You need multicast support on the network card. You may also need to enable the "all ones" broadcast address.

Multicast support is probably already built into your network card and kernel. To check, run the `ifconfig` command. You should see output for your network card(s). Just under the entries for the associated IP addresses, you should see the following:

```
UP BROADCAST RUNNING MULTICAST  MTU:1500  METRIC:1
```

If you don't see `MULTICAST` in this line, you'll need to reconfigure network support for `MULTICAST` in the kernel. Refer to Chapter 12 for details.

If you have an older client, such as Microsoft Windows 95 on your network, you'll need to configure the route to the "all ones" broadcast address, which is 255.255.255.255. To set this up on the first Ethernet card on your computer, run the following command:

```
# route add -host 255.255.255.255 dev eth0
```

The Configuration File: */etc/dhcpd.conf*

Now you can configure the main DHCP server configuration file, `/etc/dhcpd.conf`. Let's start with a sample file from the `dhcp-*` RPM, `dhcp.conf.sample` in the `/usr/share/doc/dhcp-`*`versnum`* directory. This sample lists a number of IP addresses, which you'll want to change to match the settings for your own network.

To learn more about DHCP servers, you may find it helpful to analyze the file in detail. The following is based on a line-by-line excerpt from the sample file. The first line allows Dynamic DNS

updates to the latest available "interim" standard. A number of IP addresses are shown. If the applicable IP addresses for your network are different, substitute accordingly.

```
ddns-update-style interim;
```

You may not want individual users to update their hostname or IP address entries in the DNS server, so you use this command:

```
ignore client-updates;
```

Alternatively, you can use the command `allow client-updates`, which permits users to update their hostname or IP address entries.

The following line sets the default range of allowable IP addresses. Some of these addresses may be reserved for specific computers by later commands:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
```

If your LAN is connected to another LAN, there should be a gateway IP address on a router that connects your LAN to the other. The following command specifies that gateway IP address:

```
option routers 192.168.0.1;
```

The following command is straightforward; it specifies the subnet mask, also known as the *network mask,* for the network:

```
option subnet-mask 255.255.255.0;
```

If you configure an NIS authorization database for your network, you can specify its domain (substitute it for `domain.org` in this command). For more information on NIS, see Chapter 23.

```
option nis-domain "domain.org";
```

Naturally, you probably have a domain name for your network. Based on the examples earlier in this chapter, it may be something such as `example.com`. In this command, substitute the domain name for your LAN for `domain.org`.

```
option domain-name "domain.org";
```

If you've set up a DNS server on your LAN, list its address here. It can help this DHCP server find your DNS server for updates as required. You can use similar lines to identify the servers for incoming or outgoing e-mail (`option pop-server` or `smtp-server`), a web server (`option www-server`), or even a server dedicated to log files (`option log-server`).

```
option domain-name-servers 192.168.1.1;
```

The next statement helps you keep your network synchronized. The time is shown in seconds, relative to Greenwich Mean Time (GMT). In other words, U.S. Eastern Standard Time is 18000 seconds, or 5 hours, behind GMT. If you are in a different time zone, substitute accordingly.

```
option time-offset -18000; # Eastern Standard Time
```

NOTE *In Linux, GMT is also known as UTC.*

Some computer clocks are faster than others. Computer clocks can slow down if a battery is low. If you have several computers running the same process, such as a web server, it can be important to synchronize their clocks. This is possible with a Network Time Protocol (NTP) server, which you may have configured in Chapter 13 with the `redhat-config-time` utility. You may want to set up a connection to one of these servers; for example, you can use the IP address for `clock.redhat.com`. This statement allows you to call the NTP server of your choice:

```
option ntp-servers 192.168.1.1;
```

Some Linux computers are configured as part of a Microsoft Windows–based network. One of the Microsoft name services for different computers is based on NetBIOS names. This is known as the Windows Internet Naming Service (WINS).

```
option netbios-name-servers 192.168.1.1;
```

It is possible to configure the DHCP server as a "p-node" computer; in other words, it looks for a WINS server and possibly a LMHOSTS file for name resolution.

```
option netbios-node-type 2;
```

You can configure a range of IP addresses that this DHCP server can assign to computers on remote networks. (If you're only serving the local LAN, remove the `dynamic-bootp` variable from this line.) These addresses must fit within the range of defined network addresses.

```
range dynamic-bootp 192.168.0.128 192.168.0.254
```

DHCP servers assign IP addresses on a temporary basis. The first time an IP address may be renewed is the `default-lease-time`, in seconds.

```
default-lease-time 21600
```

An IP address should be renewed by the `max-lease-time`, in seconds.

```
max-lease-time 43200
```

You can assign a fixed IP address, based on the hardware address of a specific computer's network card:

```
host ns {
     hardware ethernet 12:23:34:45:AB:CD
     fixed-address 207.175.42.254
}
```

Once you've customized this file for your LAN, save it as `/etc/dhcpd.conf`. I've illustrated what I've done for my network in Figure 19.7.

Starting the DHCP Server

To run the Linux DHCP server, you need a network card that already has an IP address. If necessary, use the `ifconfig` command to assign an IP address, as discussed in Chapter 16.

FIGURE 19.7
A private network */etc/dhcpd.conf*

```
ddns-update-style interim;
ignore client-updates;
allow booting;
allow bootp;

subnet 192.168.1.0 netmask 255.255.255.0 {

# --- default gateway
        option routers                  192.168.1.113;
        option subnet-mask              255.255.255.0;

        option domain-name              "example.com";
        option domain-name-servers      192.168.1.4;

        option time-offset              -18000; # Eastern Standard Time

        range 192.168.1.128 192.168.1.254;
        default-lease-time 21600;
        max-lease-time 43200;

        # we want the nameserver to appear at a fixed address
        host enterprise3d {
                hardware ethernet 00:40:F4:3C:05:58;
                fixed-address 192.168.1.4;
#               option host-name enterprise3d.example.com;
        }
        host enterprise3 {
                hardware ethernet 00:0C:29:1C:BB:76;
                fixed-address 192.168.1.13;
#               option host-name enterprise3.example.com;
        }
}
                                                            29,1-8        All
```

Starting the DHCP service is easy. Just run the `dhcpd` script with a command such as `service dhcpd start`. Remember to use a command such as `chkconfig` to make sure your DHCP server starts the next time you boot Linux.

DHCP Servers and Remote Networks

When you can configure a DHCP server to reserve a series of IP addresses for remote networks (see the range `dynamic-bootp` variable in the previous section), a DHCP server needs help. Normally, gateways or routers that sit between networks block DHCP messages. That is where you should implement the BOOTP protocol, which opens a path through a router or gateway for DHCP communication between your LANs.

To set up BOOTP, install the `dhcrelay` daemon (from the `dhcp-*` RPM package) on the gateway or router computer. Then you can configure command options in the `/etc/sysconfig/dhcrelay` configuration file. For example, the following commands in that file let `dhcrelay` listen on both the `eth0` and `eth1` network cards. The `DHCPSERVERS` should be connected to at least one of these network cards. You can then specify any network cards connected to networks that need remote DHCP service.

```
INTERFACES="eth0 eth1"
DHCPSERVERS="192.168.1.4"
```

Remember to start the `dhcrelay` script and use `chkconfig` to make sure that `dhcrelay` is active the next time you boot Linux.

A Lease Database

Once computers on your networks start getting addressing information from your DHCP server, the results will be documented in `/var/lib/dhcp/dhcpd.leases`. An example of this file is shown in

Figure 19.8, which displays IP address assignments to the hardware address of different network cards on your LAN.

FIGURE 19.8
dhcpd.leases

```
# All times in this file are in UTC (GMT), not your local timezone.   This is
# not a bug, so please don't ask about it.   There is no portable way to
# store leases in the local timezone, so please don't request this as a
# feature.   If this is inconvenient or confusing to you, we sincerely
# apologize.   Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-V3.0pl2

lease 192.168.1.252 {
  starts 5 2004/04/16 16:35:49;
  ends 5 2004/04/16 22:35:49;
  binding state active;
  next binding state free;
  hardware ethernet 00:0c:29:1c:bb:76;
}
lease 192.168.1.254 {
  starts 5 2004/04/16 16:37:32;
  ends 5 2004/04/16 22:37:32;
  binding state active;
  next binding state free;
  hardware ethernet 00:40:f4:3c:05:58;
}
~
```

Working with DHCP and BOOTP Clients

Configuring a DHCP client is fairly easy. Once networking is configured, you need to point the startup script for your network card to look for a DHCP server. Once configured, your computer broadcasts a request looking for a DHCP server the next time it boots.

Applicable */etc/sysconfig* Files

Naturally, you need to make sure that networking is enabled. Check your `/etc/sysconfig/network` file. It should include the following entry:

```
NETWORKING=yes
```

Now revise your network card configuration file. It's usually in the `/etc/sysconfig/network-scripts` directory. If the network card is `eth0`, the filename is `ifcfg-eth0`, and the file should contain the following:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

There are two alternatives for the `BOOTPROTO` variable: `bootp` and `dialup`. These alternatives are almost self-explanatory; `bootp` assumes the DHCP server is on a remote network, and `dialup` configures the device for a dial-up connection, such as to an ISP on the Internet.

dhclient

Once the configuration files are changed, the easiest way to start your computer as a new DHCP client is with the `dhclient` command. The result should resemble that shown in Figure 19.9.

NOTE *Red Hat has changed the name of its DHCP client a number of times in the past couple of years; previous names included* `dhcpcd` *and* `pump`.

FIGURE 19.9
Leasing an IP address

```
[root@Enterprise3d root]# dhclient
Internet Software Consortium DHCP Client V3.0pl2
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP

Listening on LPF/lo/
Sending on   LPF/lo/
Listening on LPF/eth0/00:40:f4:3c:05:58
Sending on   LPF/eth0/00:40:f4:3c:05:58
Sending on   Socket/fallback
DHCPDISCOVER on lo to 255.255.255.255 port 67 interval 5
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.254
bound to 192.168.1.254 -- renewal in 8825 seconds.
[root@Enterprise3d root]# 
```

Summary

There are two key services that help your Linux computer communicate on a TCP/IP network such as the Internet: DNS and DHCP. This chapter showed you how to configure clients and servers for each service.

A DNS server is a database of FQDN and IP addresses. You can configure master, slave, caching-only, or forwarding DNS servers. It's best to configure DNS directly, as the Red Hat GUI tool available as of this writing is less than perfect. The main DNS configuration files are `/etc/named.conf` and several files in `/var/named`. If you use the Red Hat GUI tool, add special configuration options to `/etc/named.custom`. Once you've configured the server, you can start the `named` daemon, which controls DNS, with the `service named start` command.

Generally, no special configuration is required to set up a DNS client. Normally, a DNS client will search through `/etc/hosts` before moving to the DNS servers identified in `/etc/resolv.conf`.

A DHCP server enables you to manage the IP addresses on your network. You can also set up other basic network information in the `/etc/dhcpd.conf` configuration file, such as gateways, DNS servers, NIS servers, and even SMTP servers. As long as the DHCP server computer has a network card with an IP address, you can start the DHCP server with the `service named dhcpd` command. Configuring a gateway computer to transfer DHCP messages between networks is possible with the `dhcrelay` daemon. Once you've set up a DHCP server, you can lease an address with the `dhclient` command. Leased addresses are stored in a `/var/lib/dhcp/dhcpd.leases` database.

Configuring DHCP clients is fairly easy; the key file is the configuration file for your network card in the `/etc/sysconfig/network-scripts` directory. If there is a DHCP server for your LAN, you can get your IP addressing information for it immediately with the `dhclient` command.

In the next chapter, we'll look at the major print system for Red Hat Enterprise Linux: the Common Unix Printing System.

Chapter 20

Printing with CUPS

WHEN YOU INSTALL RED Hat Enterprise Linux, it does not automatically detect printers. Therefore, all administrators need to know some of the arcane details of printer configuration.

Red Hat Enterprise Linux includes the Common Unix Printing System (CUPS) by default. CUPS, which is based on Internet Printing Protocol (IPP) version 1.1, allows administrators to organize networked printers in groups. The CUPS technical term for a group of printers is a *class*. Red Hat's GUI Printer Configuration tool works well with CUPS printers.

In the enterprise, you can also organize large numbers of printers through the CUPS web-based interface. For example, you can set up a class of printers as if it was a single printer. Print jobs go to the first available printer in that class.

You should also understand the contents of the associated configuration files. While the language in the `/etc/cups` configuration files may seem ancient, it is quite similar to the language associated with the Apache web server configuration file in Chapter 25.

NOTE *The alternative LPRng (Line Printer, Next Generation) print system is no longer included with Red Hat distributions.*

If you're more familiar with LPRng or LPD, you can still use many of the associated features. CUPS includes an `xinetd` service (`cups-lpd`) that lets you use standard LPD commands such as `lpr` and `lpq`. This chapter covers the following topics:

- Using the Internet Printing Protocol
- Red Hat's Printer Configuration tool
- Configuring the Common Unix Printing System (CUPS)

Using the Internet Printing Protocol

In the past, Unix and allied systems such as Linux did not do a very consistent job with printer interfaces. As companies such as AT&T, HP, and Sun created their own versions of Unix, they also created proprietary print interfaces. While Linux did well to adapt the LPD packages, the evolving industry standard is based on the Internet Printing Protocol (IPP).

CUPS is the Linux and Unix way of working with IPP. It was developed by Novell and Xerox with four goals in mind—to enable users to:

- Find available printers on a network
- Send print jobs to an IPP-configured printer
- Read the status of their print jobs
- Cancel any print jobs they may have created

CUPS allows you to send print jobs to a specific URI, such as `parallel:/dev/lp0`.

NOTE *A URI is a Uniform Resource Identifier. You're probably more familiar with the URL (Uniform Resource Locator), which is a subset of an URI. As you know, a URL is used in web browsers to point to sites such as* `ftp://ftp.redhat.com` *or* `http://www.sybex.com`*. A URI can point to more things, such as* `mailto:abc@def.ghi`*,* `smb://comp1/printername`*, or* `parallel:/dev/lp1`*.*

CUPS implements IPP in a number of different ways. Several of the standards, as described in Table 20.1, probably seem familiar to those of you who know LPD. The standard actions shown are far from a comprehensive list. More detailed information is available from the developers of CUPS, Easy Software Products, at `www.easysw.com`.

TABLE 20.1: CUPS FUNCTIONALITY

ACTION	**DESCRIPTION**
Print	Sends a file to a printer at a specific URI
Validate	Makes sure that a job has the right priority, printer, and so on
Create	Sets up an empty print job
Send	Sends a file for processing as a print job
Cancel	Cancels a print job
Pause	Stops action by a printer
Resume	Resumes action by a printer
Purge	Clears jobs from a printer's spool

In addition, CUPS includes a number of administrative functions over and above the standard LPD system. Some of these functions are shown in Table 20.2. Once again, this is not a comprehensive list.

With these basic concepts in mind, you're ready to learn how to configure CUPS on your computer and network.

TABLE 20.2: SPECIAL CUPS FUNCTIONS

ACTION	DESCRIPTION
CUPS-Get-Default	Finds the URI for the default printer
CUPS-Get-Printers	Finds the URIs for all printers configured on the network with CUPS
CUPS-Add-Modify-Printers	Adds or modifies a printer through CUPS
CUPS-Delete-Printer	Deletes a printer from a CUPS class
CUPS-Get-Classes	Finds the types of printers available in each CUPS class
CUPS-Add-Modify-Class	Adds a new printer class or modifies an existing CUPS printer class
CUPS-Delete-Class	Deletes an existing class of CUPS printers
CUPS-Accept-Jobs	Sets a specific printer or print class to start accepting print jobs
CUPS-Reject-Jobs	Sets a specific printer or print class to start rejecting print jobs

Red Hat's Printer Configuration Tool

Because the commands associated with `/etc/cups/cupsd.conf` configuration files are so obscure, the more popular option for configuring CUPS printers is a graphical tool. Red Hat's Printer Configuration tool is based on the `redhat-config-printer` and `redhat-config-printer-gui` RPMs. You can run this command from a text or a GUI command-line console. While the look and feel is different (see Figures 20.1 and 20.2), the information is the same.

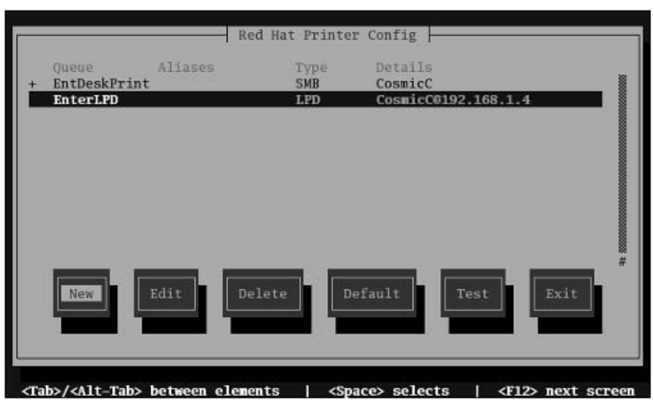
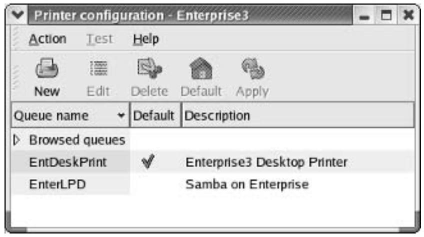


FIGURE 20.1
Printer Configuration tool, command-line version

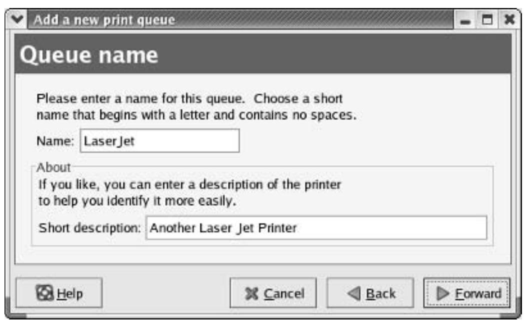
FIGURE 20.2
Printer Configuration tool, GUI version



It's easy to use `redhat-config-printer` to set up a new printer. The following steps are based on the GUI version of this tool, which starts in a Printer Configuration window:

1. Click New; when you see the Add A New Print Queue dialog box, click Forward to continue.
2. In the Queue Name dialog box, enter a short name for your printer and a description, similar to what is shown in Figure 20.3. When you're ready, click Forward to continue.

FIGURE 20.3
Adding a printer queue



3. In the Queue Type dialog box, you'll see available printer ports, similar to what's shown in Figure 20.4. For example, `/dev/lp0` corresponds to the first parallel port. If you're configuring a local printer, select the available port of your choice and then click Forward. If that does not meet your needs, try one of the following alternatives:
   - If you don't see your port, first try the Rescan Devices option.
   - If that does not work, you can click Custom Device and enter the device associated with your printer port.
   - If you want to select a network printer, click the Select A Queue Type drop-down text box. Available choices are listed in Table 20.3. You'll be prompted to specify the appropriate network settings for the remote printer.

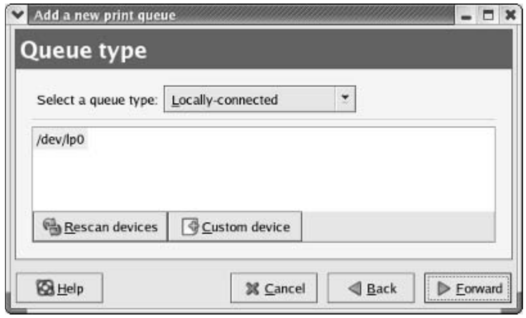


FIGURE 20.4
Selecting a queue type

TABLE 20.3: NETWORK QUEUE TYPES

TYPE	DESCRIPTION
Networked CUPS (IPP)	For printers on a remote CUPS server; requires the server host or domain name and IPP path.
Networked Unix (LPD)	For printers on a remote LPD server; requires the server host or domain name as well as the name of the remote queue.
Networked Windows (SMB)	For printers on a remote Microsoft Windows print server; should detect shared printers automatically.
Networked Novell (NCP)	For printers on a remote Novell print server; requires the server host or domain name, the queue name, and the authorized username and password.
Networked JetDirect	For printers on a remote JetDirect print server that's directly connected to the network; requires the name of the JetDirect printer.

4. In the Printer Model dialog box, select the make and manufacturer for your printer. Select the driver best suited to your printer, using these guidelines, and then click Forward to continue:
   - Some printers work with the PostScript print driver, and printers that process raw print data can use the Raw Print Queue driver; generic drivers are also available for text and various dot-matrix printers.
   - To select a specific model, click the Generic (Click To Select Manufacturer) text box; a series of manufacturers display, similar to Figure 20.5. Once you make your selection, you'll be able to select a print model that most closely matches your printer.
5. You'll now see the Finish, And Create The New Print Queue dialog box. It will include a summary of your selections, similar to Figure 20.6. If you're satisfied with your selections, click Apply.

FIGURE 20.5
Selecting a printer manufacturer

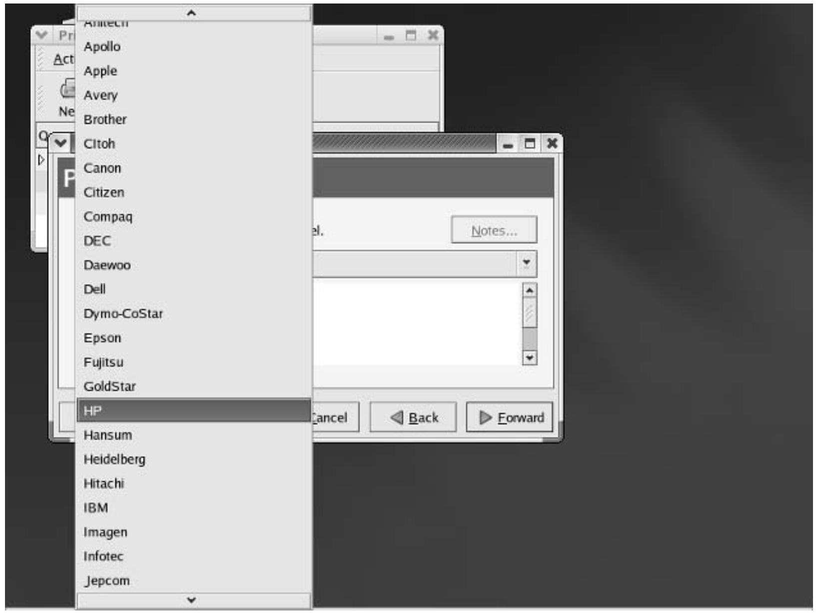
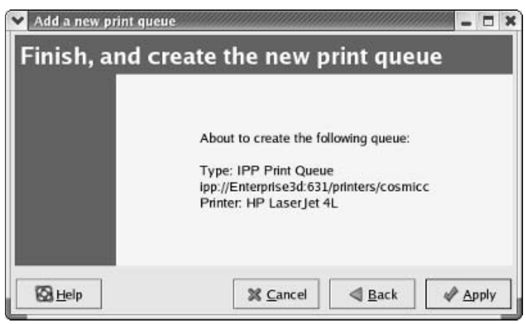


FIGURE 20.6
Print configuration summary



6. You're given an opportunity to print a test page. It's a good idea; if you're connected to your printer, click Yes.

You're taken back to the Printer Configuration dialog box. You should now see an entry for your new printer. You can edit the settings; simply highlight the printer, and click Edit. This opens the Edit A Print Queue dialog box for the printer you just configured. The five tabs in this dialog box are summarized in Table 20.4.

Before leaving the Red Hat Printer Configuration tool, be sure to click Apply. This action writes your changes to `/etc/cups/cupsd.conf` and then restarts the `cupsd` print daemon

Table 20.4: Editing Printer Settings

Tab	Description
Queue Name	Lets you specify the name of the print queue
Queue Type	Allows you to revise the device, even to a networked printer
Queue Options	Lets you configure basic settings for banner pages, margins, and the filter
Printer Driver	Lets you change the driver
Driver Options	Allows you to specify more driver settings

Configuring the Common Unix Printing System

In many cases, configuring the Common Unix Printing System (CUPS) is easy. Since CUPS is the default, if the right packages are installed CUPS may already be activated on your computer. Many LPD commands can be used on CUPS printers; all you need to do is activate the `xinetd`-managed daemon, `cups-lpd`.

You can configure many CUPS printers through a web-based interface on TCP/IP port 631, which is the communications channel for IPP. However, if you're configuring a group of CUPS printers, you need to know how to directly edit the CUPS configuration files in the `/etc/cups` directory.

Check your current CUPS RPM packages. Install them if they're not already on your computer. These packages are summarized in Table 20.5.

Table 20.5: CUPS RPM Packages

Package	Description
`cups-*`	The main CUPS package, which includes basic commands and default configuration files.
`cups-libs-*`	A package that allows you to use CUPS commands without having to use LPD commands such as `lpr`.
`cups-devel-*`	The CUPS development libraries.
`foomatic-*`	A spooler independent database of printers; the Red Hat version of this RPM is different from other Linux distributions; it's designed for the Red Hat Printer Configuration tool.
`gimp-print-cups-*`	Another series of print drivers, usable for more than The GIMP; for more information, see `gimp-print.sourceforge.net`.
`hpijs-*`	Print drivers optimized for HP printers.

In the sections that follow, we start with the web-based interface and then offer a detailed look at each of the CUPS configuration files in `/etc/cups`. Finally, we look at some basic CUPS commands and the `cups-lpd` service that lets you use LPD commands.

NOTE *The names of the CUPS files, daemons, and scripts may be a bit confusing. The CUPS daemon is* `cupsd`*, in the* `/usr/sbin` *directory. However, Red Hat Enterprise Linux lets you start and stop CUPS with a* `cups` *script in the* `/etc/rc.d/init.d` *directory. Finally, the main CUPS configuration file is* `cupsd.conf`*, in the* `/etc/cups` *directory.*

Web-based Configuration

You can set up CUPS printers on the web browser of your choice. As the Printing Support package group is installed by default, the CUPS RPM packages are probably already installed, and the `cups` daemon should be active. In that case, all you need to do is open the local browser of your choice on TCP/IP port 631.

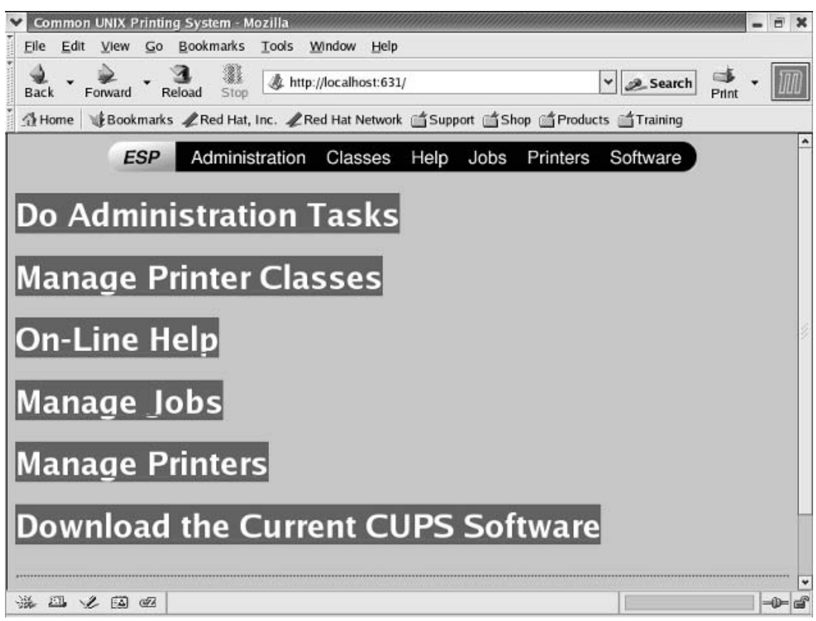
However, the version of this tool included with Red Hat Enterprise Linux is functionally limited. The `foomatic` RPM on other Linux distributions provides a diverse library of printer drivers. The Red Hat version of this package is designed to work only with the Red Hat Printer Configuration tool. If you configure printers on Red Hat Enterprise Linux with the web-based tool, you probably won't find a suitable driver. Therefore, we do not cover this process in this book.

But the web-based tool is still important. It provides a means to create a print class. A CUPS print class is a group of two or more printers. Once you create a print class, you can print to that class as if it were any other printer. CUPS directs the print job to the first available printer in that class.

NOTE *You can run the CUPS configuration program from a web browser on a remote computer. However, this requires you to have no firewall between those two computers—at least none that block port 631. While we don't encourage this practice, you may find the risks acceptable if you're on a LAN protected from outside networks with a firewall.*

Now open the browser of your choice, and direct it to `http://localhost:631`. Figure 20.7 shows the result in the Mozilla web browser.

FIGURE 20.7 The CUPS printer configurator



TIP *You may see the following message in your browser: "The connection was refused when attempting to contact* `servername`*:631." If you do, you haven't activated the* `cupsd` *daemon, or you have a firewall that's blocking access to port 631.*

As you can see, there are six different command options; the ESP link at the top of the web page is a link to the people behind CUPS, Easy Software Products at `www.easysw.com`. The other options are fairly straightforward and are summarized in Table 20.4.

TABLE 20.6: CUPS CONFIGURATION MENU OPTIONS

OPTION	**DESCRIPTION**
ESP	Navigates to `www.easysw.com`
Administration: Do Administration Tasks	Allows you to add or manage printers, classes, and print jobs
Classes: Manage Printer Classes	Lets you add or manage a group of printers as a class
Help: On-Line Help	Includes HTML and PDF manuals related to CUPS
Jobs: Manage Jobs	Allows you to manage current print jobs in the CUPS system
Printers: Manage Printers	Lets you add or manage an individual printer
Software: Download The Current CUPS Software	Navigates to `www.cups.org` for the latest available CUPS packages

Since the Administration link provides an "all-in-one" configuration interface, we'll examine these options (except ESP) in reverse order.

TIP *Before you continue, back up the files in your* `/etc/cups` *directory. The original format of these files will be used later in this chapter.*

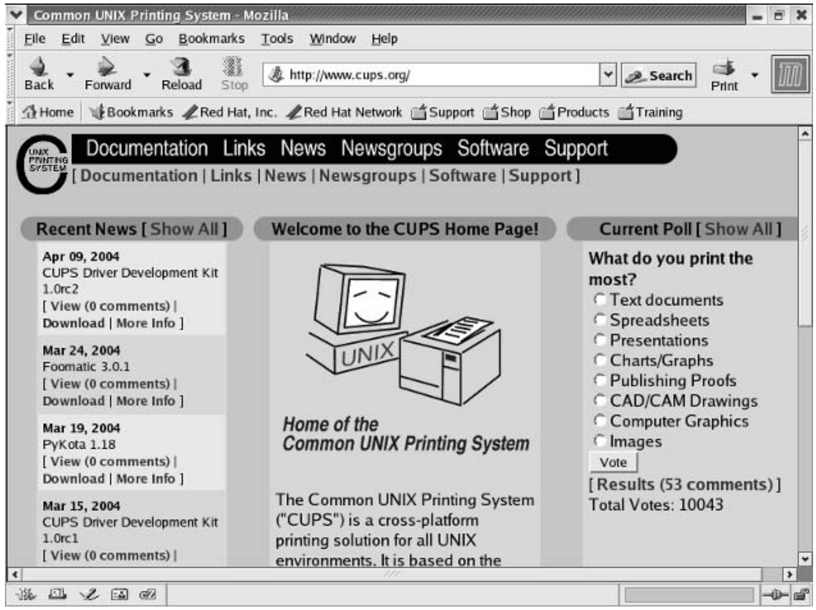
You can use the Red Hat Printer Configuration tool described earlier in this chapter to configure or edit the printers of your choice. As printer drivers on Red Hat Enterprise Linux are limited for the web-based tool, we do not examine that process in this book. However, we do examine all the other tools available through the web-based tool.

DOWNLOADING CUPS

If you want to download the latest version of CUPS, it's available from the CUPS website at `www.cups.org`; see Figure 20.8. As of this writing, downloadable versions from `www.cups.org` are available only in tarball-style formats and cannot be customized for Red Hat Enterprise Linux. Naturally, Red Hat does not support these packages.

NOTE *The* `www.cups.org` *website is maintained by Easy Software Products; their home page is* `www.easysw.com`*. But remember, CUPS is open-source software licensed under the GPL.*

FIGURE 20.8
The CUPS home page



Therefore, it's usually best to download the latest version of CUPS through the Red Hat Network.

Managing Jobs

It is easy to check the current queue of print jobs. Navigate to `http://localhost:631` to return to the main CUPS menu. Click the Jobs or Manage Jobs link, and you'll see a current list of jobs in the queue. These jobs are stored in files in the `/var/spool/cups` directory. If any jobs are pending, you'll see them in a format similar to what is shown in Figure 20.9.

As shown in the figure, it's easy to Hold or Cancel pending print jobs. Any job that is held is stored in `/var/spool/cups`; other jobs are processed first. You can then release the job to the queue as desired. More details on each job are available by clicking the associated ID.

One useful CUPS feature is a history of completed jobs. Click the Show Completed Jobs button to inspect your completed jobs, similar to what's shown in Figure 20.10. You can use this feature to monitor the activity of your printers to see if a print job is complete.

FIGURE 20.9
Pending CUPS print jobs

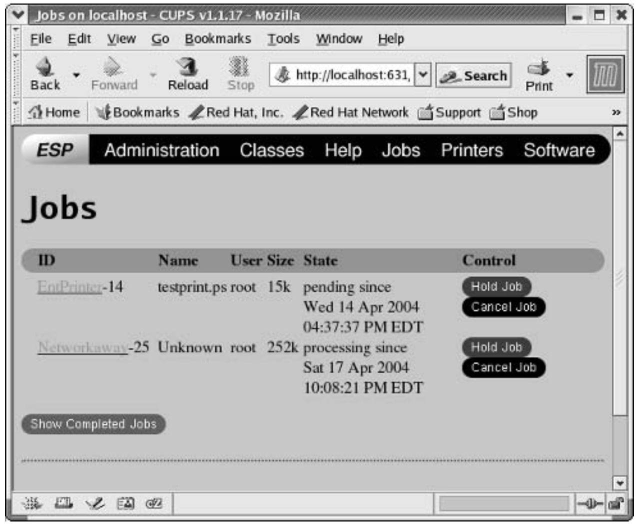


FIGURE 20.10
CUPS completed print jobs

Accessing Online Help

Considerable online help is available for CUPS. All you need to do is click Help or On-Line Help. Either link opens the CUPS documents that are installed with the `cups-*` RPM in your local computer. Briefly, they include the documents described in Table 20.7. Additional manuals are available for CUPS developers.

Table 20.7: CUPS Online Documents

Document	Description
An overview of the Common Unix Printing System	Describes the basic structure of CUPS, how it works with IPP 1.1, and compatibility with LPD commands
Software Users Manual	Includes a detailed description of the way you can customize printing with the right CUPS commands
Software Administrators Manual	Includes a detailed description of the CUPS installation and the language of the `/etc/cups` configuration files
CUPS Implementation of IPP	Compares CUPS functionality to IPP requirements

Now navigate to `http://localhost:631` to return to the main CUPS menu.

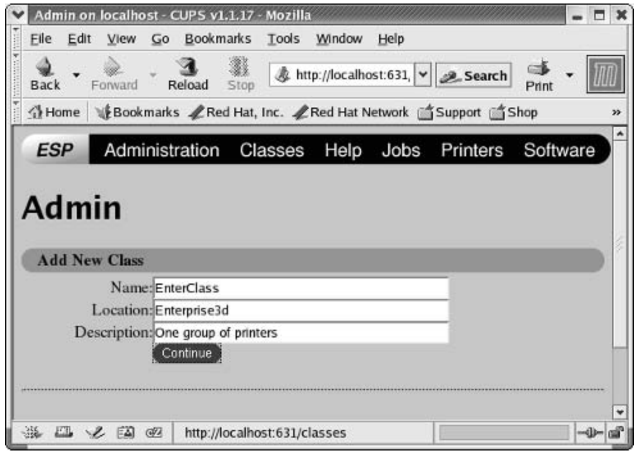
Managing Printer Classes

One of the strengths of CUPS is how it allows you to organize groups of printers. Once you've configured your printers, you can group them into CUPS classes. When you send a print job to a class, the first available printer in that class processes the job. Users no longer need to wait until a specific printer is free. In the CUPS menu, click Classes. CUPS takes you to a screen with currently configured printer classes. Even if you're logged in as the root user, CUPS should prompt you for your administrative account, as shown in Figure 20.11. (We're assuming this is the first time you've requested administrative functionality in the web-based tool.)

Once you've entered the appropriate username (usually root) and password, you're taken to the Add New Class screen, shown in Figure 20.12.

Figure 20.11
Authorized access

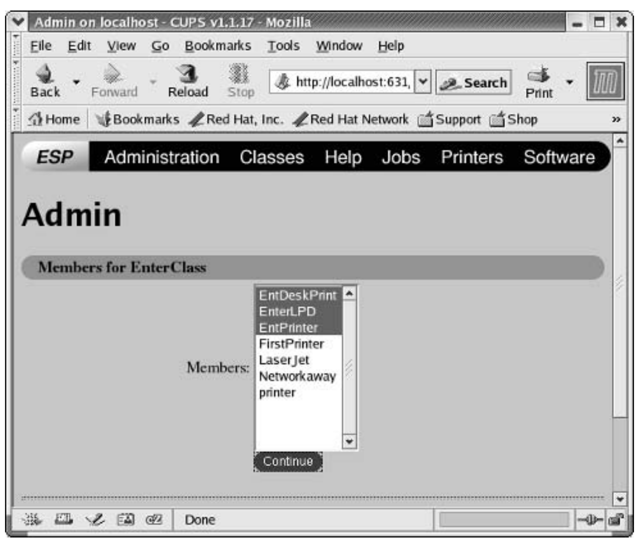
FIGURE 20.12
Adding a new printer class



In this case, the new class name is EnterClass, which is different from any existing printer name. The Location and Description fields are essentially the same as when you added a new CUPS printer; Location corresponds to the hostname or domain name associated with the print server, and Description gives you a chance to add a descriptive comment about the new printer class.

Click Continue; CUPS now takes you to the Members For *PrintClassName* screen. All configured CUPS printers are included in this screen, even if they're already assigned to a different class. To add the printers shown in Figure 20.13 to the new EnterClass class, highlight them and click Continue. CUPS displays a message that the EnterClass class has been added successfully. Now you can print to EnterClass, and CUPS will send the job to the first available printer in that class.

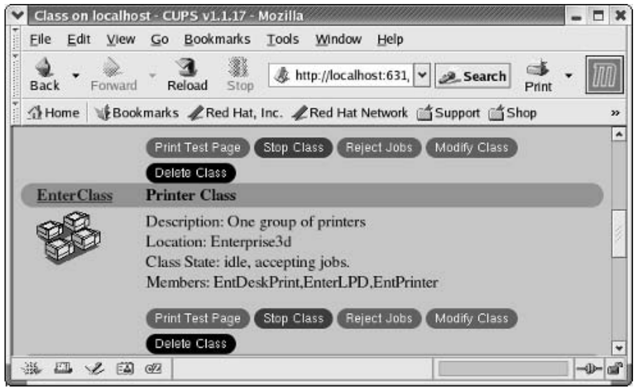
FIGURE 20.13
Adding printers to the new class



Click Classes again, and you'll see a screen with your configured printer classes. Figure 20.14 illustrates the class that we created, with the members EntDeskPrint, EnterLPD, and EntPrinter.

Now navigate to `http://localhost:631` to return to the main CUPS menu.

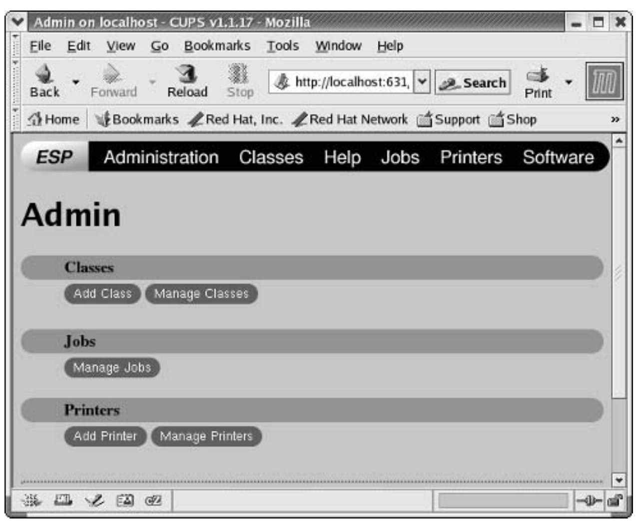
FIGURE 20.14
A defined printer class



ADMINISTRATIVE TASKS

When you click Administration or Do Administration Tasks, you're taken to a menu where you can manage printer classes, print jobs, and printers. As shown in Figure 20.15, this is close to an "all-in-one" CUPS administration menu.

FIGURE 20.15
The CUPS Administration menu



The *lpadmin* Command

While it's common for expert Linux administrators to administer from the command-line interface, many have come to trust the CUPS web-based configurator. In general, Linux administrators don't trust the extra layer associated with a GUI interface; more can go wrong. Not surprisingly, it's still possible to administer CUPS printers from the command line by using the `lpadmin` command. So many printer types and models are available, however, that this command becomes impractical.

But you can administer from the command line. One key function is to set up a user-based quota for your printer. This can help you track usage. For example, you can set quotas on a specific printer using the `lpadmin` command. The following command specifies that all users are limited to 10 pages per day on the printer named MyLaserJet:

```
# lpadmin -p MyLaserJet -o job-quota-period=86400 -o job-page-limit=10
```

Alternatively, you could use the `-o job-k-limit` switch to limit the amount of data sent to the printer in kilobytes.

You can also limit access to a specified printer. For example, the following command limits access to the printer MyLaserJet to users ez and cchavez:

```
# lpadmin -p MyLaserJet -u allow:ez,cchavez
```

Alternatively, this command prohibits access to the printer MyLaserJet for the user mj:

```
# lpadmin -p MyLaserJet -u deny:mj
```

The `lpadmin` command affects the data in `/etc/cups/printers.conf`.

The *lpstat* Command

You can check the status of your printers and classes with the `lpstat` command. It's fairly straightforward; the `-c` *`class`* option lists members of the specified class; the `-v` *`printer`* option lists the device or address for the specified printer.

Configuration Files

The CUPS configuration files are stored in the `/etc/cups` directory. If you're familiar with the Apache web server described in Chapter 25, you should be comfortable with CUPS.

The language is similar. Remember, CUPS lists printers by their URIs, such as `ipp://Enterprise3d/MyLaserJet`. As you know, URLs list locations with addresses such as `http://www.sybex.com`. The standard configuration files are listed in Table 20.6; we examine `/etc/cups/cupsd.conf` in detail in the following section.

TABLE 20.8: CUPS CONFIGURATION FILES (IN */ETC/CUPS*)

FILE	DESCRIPTION
`classes.conf`	Specifies different groups of printers; when you create a new printer class with the CUPS web-based tool, the details are written here.

Continued on next page

TABLE 20.8: CUPS CONFIGURATION FILES (IN */ETC/CUPS*) *(continued)*

File	Description
`client.conf`	Points to a default CUPS server; you may specify encryption requirements.
`cupsd.conf`	The main CUPS configuration file.
`lpoptions`	Allows you to set, save, and view printer options
`mime.convs`	Lists filters for various file formats, such as documents and images.
`mime.types`	Lists file types that can be processed through CUPS printers.
`printers.conf`	The configuration file changed by the CUPS web-based tool; the details are written here.
`pstoraster.convs`	Contains a conversion filter for GhostScript files, the way GNU works with PostScript printers.

/etc/cups/cupsd.conf

While you can set up CUPS printers and classes with the web-based tool, to administer a group of printers you need to understand the main CUPS configuration file, `/etc/cups/cupsd.conf`. This section explains the default version of this file in detail; as you'll see, a number of variables are commented out that you can activate for your network of printers.

The variables listed in this section don't exactly match the order shown in the default `/etc/cups/cupsd.conf` configuration file; for example, variables related to log files are grouped together in their own section.

Other variables are available for `cupsd.conf`; for more information, see the CUPS Software Administrator's Manual, available in the On-Line Help section of the CUPS GUI configuration program.

NOTE *Remember, the* `#` *is a comment character; you need to remove it to activate the command. In some cases, the command shown as a comment is the default.*

The settings (other than defaults) that you configure with the Red Hat Printer Configuration tool are added to the end of this file. If you're unfamiliar with CUPS, one good learning experience is to analyze the commands associated with a new local and a network printer that you may configure.

Server Identity

The `ServerName` variable is straightforward; it lists the visible name of your CUPS print server computer. By default, it is set to the hostname of the local computer.

```
#ServerName myhost.domain.com
```

This name should match the `ServerName` variable on CUPS client computers in `/etc/cups/client.conf`. Next, the `ServerAdmin` variable is essentially set to the e-mail address of the "webmaster" of the CUPS server.

```
#ServerAdmin root@your.domain.com
```

Standard Directories

Several files are listed in `cupsd.conf`; if listed with the relative path, they are relative to the directory listed as `ServerRoot`; by default, this is set to `/etc/cups`:

```
#ServerRoot /etc/cups
```

By default, the CUPS RPM packages store standard print data in the `/usr/share/cups` directory. This includes classifications, fonts, character sets, the help documents, and more. You can change where CUPS looks for this directory by changing the following variable:

```
#DataDir /usr/share/cups
```

When you send a print job, it is processed into a file that is stored on a spool. Normally, the file stays in the spool directory until the printer physically processes the job. The standard directory is specified with the `RequestRoot` variable. By default, it's `/var/spool/cups`:

```
# RequestRoot /var/spool/cups
```

CUPS also needs a temporary directory writeable by all users. Filters may be stored in this directory while a print job is being processed. While the default is `/var/tmp`, Red Hat Enterprise Linux configures this in the `/var/spool/cups/tmp` directory, as shown here:

```
#TempDir /var/spool/cups/tmp
```

If you create your own temporary CUPS directory as root, you can set the appropriate permissions with this command:

```
# chmod a+t /tempdir
```

To help you visualize the result, here is the output from an `ls -l /var/spool/cups` command:

```
drwx------T   2 lp   sys    4096 Mar 3 12:48 tmp
```

Log File Variables

As described in Chapter 13, most log files are stored in the `/var/log` directory. CUPS log files are no exception; they are stored in the `/var/log/cups` directory. The standard log file lines are as follows:

```
#AccessLog /var/log/cups/access_log
#ErrorLog /var/log/cups/error_log
#PageLog /var/log/cups/page_log
```

These variables are set to default values. Of course, you can redirect these log files to the directory of your choice. These logs collect data as described in Table 20.9.

Table 20.9: CUPS Log Files

File	Description
`access_log`	Lists HTTP files accessed through the CUPS web management tool.

Continued on next page

Table 20.9: CUPS Log Files *(continued)*

File	Description
`error_log`	Includes more than just error messages; in standard log format, includes err, warn, info, and debug messages.
`page_log`	Notes each page that is sent to a printer.

Chapter 13 described how log files are rotated on a weekly basis. The `MaxLogSize` variable also forces the aforementioned logs to be rotated once the log file reaches a certain size. If the variable is not set, the default is 1MB; if it's set to 0, logs aren't rotated unless specified by another job such as those listed in the `/etc/cron.daily` directory.

```
MaxLogSize 0
```

Chapter 13 also described how logs collect data based on settings in the `/etc/syslog.conf` configuration file. The available levels for CUPS, which are slightly different, appear in Table 20.10. By default, `LogLevel` is set to `info`.

```
LogLevel info
```

Table 20.10: CUPS Log Levels

Level	Description
`emerg`	Conditions that prevent CUPS from working
`alert`	Items that must be addressed immediately
`crit`	Critical errors that may not prevent CUPS from working
`error`	General errors
`warn`	Warning messages
`notice`	Temporary errors
`info`	All requests and CUPS changes in status
`debug`	Basic debug information
`debug2`	All debugging information

Security Printouts

You can set a header on each printed page. If security requirements are associated with printouts on your network, you can uncomment one of the following commands:

```
#Classification classified
#Classification confidential
#Classification secret
```

```
#Classification topsecret
#Classification unclassified
```

By default, there is no `Classification`. But if there is one, the `ClassifyOverride` variable may apply. If you set this variable to `on`, it allows users to change the classification associated with a specific print job. The `ClassifyOverride` default is `off`, as shown here:

```
#ClassifyOverride off
```

The standard font used by the CUPS web-based configuration tool is set by the `DefaultCharset` variable. Common options include `iso-8859-1` and `windows-1251`. But this does not apply if a `DefaultLanguage` variable is present, or if the CUPS client sets a different `DefaultCharset`:

```
#DefaultCharset utf-8
```

The `DefaultLanguage` specifies the language used for connections to the CUPS web browser tool. By default, it's English (en); alternatives include German (de), Spanish (es), French (fr), and Italian (it).

```
#DefaultLanguage en
```

As with Apache, the `DocumentRoot` variable specifies the base directory for different HTML pages. In this case, these HTML pages are associated with the CUPS web browser tool. By default, it's set to the `/usr/share/doc/cups-`*`versionnumber`* directory.

```
#DocumentRoot /usr/share/doc/cups-versionnumber
```

Linux generally implements PostScript files using GhostScript. When such files are sent to a printer, they need the fonts as currently specified by the `FontPath` variable. By default, this variable is set as follows:

```
#FontPath /usr/share/cups/fonts
```

Print Job Management

There are four basic variables relate to how print jobs are managed. For example, you can configure your CUPS print server to keep a record of past jobs or even the spool files. The `PreserveJobHistory` variable, which is set to `Yes` by default, keeps a record of past jobs.

```
#PreserveJobHistory Yes
```

You can keep a history of past job spool files. If this variable is set to `Yes`, you can reprint previous jobs until you purge them. However, the `PreserveJobFiles` variable by default is set to `No`.

```
#PreserveJobFiles No
```

You may not have unlimited hard disk space. The `MaxJobs` variable sets a limit on the number of previous print jobs you may preserve. The default is 500.

```
#MaxJobs 500
```

Naturally, this goes hand in hand with a limit on copies, as defined by the `MaxCopies` variable.

```
#MaxCopies 100
```

Normally, it's a good idea to set quotas to track usage of your CUPS printers, as described earlier with the `lpadmin` command. Print jobs are normally not purged, so data associated with printer usage remains on your system.

Conversely, if you have not set quotas, you have no need to keep track of the number of print jobs run by any user.

You can then activate the `AutoPurgeJobs` variable, which automatically deletes print jobs from the system.

```
#AutoPurgeJobs No
```

You can configure a list of available printers in a standard file such as `/etc/printcap` with this straightforward command:

```
#Printcap /etc/printcap
```

Normally, `/etc/printcap` is based on the LPD system, developed for BSD. However, a similar format is available for the Solaris operating system. While the BSD-style system is the default, you can activate either with one of the following commands:

```
#PrintcapFormat BSD
#PrintcapFormat Solaris
```

Note *Don't worry about the* `PrintcapGUI` *variable; it's used for printer control only for the SGI IRIX operating system.*

Some print jobs need help from a program; these programs are normally stored in executable format in `/usr/lib/cups`, as specified by the `ServerBin` variable.

```
#ServerBin /usr/lib/cups
```

Most printers are configured to print graphics in *Raster* mode, dot by dot. However, the Raster Image Processing Cache variable, `RIPCache`, is used by specialized print filters such as imagetoraster and pstoraster. By default, the cache is 8MB; you can set caches in kilobytes and gigabytes with values such as `100k` or `1g`.

```
#RIPCache 8m
```

Note *In this case, RIP has nothing to do with the TCP/IP Routing Information Protocol.*

If you find that the print jobs are taxing the capacity of your server, you may want to set a `FilterLimit`. Normally, this variable is set to 0, which corresponds to no limit:

```
#FilterLimit 0
```

The number you use will be based on trial and error; a couple of guidelines are available. If you want to print to a regular printer, you should set this value to 200; if you have several regular printers, set this value higher. If you set this value lower than 200, you effectively limit CUPS to processing one job at a time.

Encryption Support

Sometimes network communication is encrypted. You can configure CUPS to read encrypted print requests. The SSL certificate and key are defined by the following variables:

```
#ServerCertificate /etc/cups/ssl/server.crt
#ServerKey /etc/cups/ssl/server.key
```

And these certificates must be refreshed over a network periodically, as driven by the `RootCertDuration` variable, in seconds.

```
#RootCertDuration 300
```

CUPS Accounts

While CUPS is started by the root user, CUPS jobs are normally run by other users with less access. And when you access CUPS from a different computer, CUPS assigns you a different username, remroot, as specified by the `RemoteRoot` variable:

```
#RemoteRoot remroot
```

The standard CUPS user is lp (yes, from the now obsolete LPRng service), and the standard group is sys, as defined by the User and Group variables. You can supersede these with the `RunAsUser Yes` command.

```
#User lp
#Group sys
```

Basic Network Settings

CUPS was developed for TCP/IP networks. When you configure CUPS, you can set it to listen for specific computers and/or IP addresses on specific ports. For example, the following commands set CUPS to tune into port 631, to listen for requests from the computer named `linux.mommabears.com`, and to listen for requests from the 192.168.22.0 network:

```
Port 631
Listen linux.mommabears.com
Listen 192.168.22.0
```

If you want to listen for a specific hostname, you need to set `HostNameLookups on`. You can even combine some of these settings; for example, the following commands set CUPS to listen for requests from the 10.11.12.0 network, on port 80:

```
Listen 10.11.12.0:80
```

NOTE *In Apache 2.0.x, the* `Listen` *directive has replaced the* `Port` *directive. See Chapter 25 for more information.*

Normally, you should stick with IP addresses in the `cupsd.conf` configuration file. Looking up domain names in a DNS server can take time and slow down your CUPS print server. However, if you set `HostNameLookup on`, CUPS uses your DNS server to look for the IP address associated with a domain name.

CUPS normally keeps open connections with web browsers, courtesy of the `KeepAlive On` variable. However, if you're administering CUPS through an older web browser such as Netscape 2.*x*, `KeepAlive` doesn't work. In that case, you need to set a time that CUPS will wait for data from the web-based tool. That's defined by the `KeepAliveTimeout` setting, which keeps the connection open for the noted period of time, in seconds.

```
#KeepAlive On
#KeepAliveTimeout 60
```

User Limits

When you set up a print server on a network, any user may request access at any time. The `MaxClients` variable limits the number of users that connect to your CUPS print server; the default limit is 100 users.

```
#MaxClients 100
```

You can log into a single host computer multiple times. That is limited by the `MaxClientsPerHost` variable.

```
#MaxClientsPerHost 0
```

You may also want to regulate the size of jobs sent through your CUPS print server. You may want very large jobs to be sent to other servers. You can set a limit with the `MaxRequestSize` variable in bytes or megabytes. However, the default is to avoid a limit by using the following command:

```
#MaxRequestSize 0
```

Related variables include `MaxJobsPerPrinter` and `MaxJobsPerUser`. If you want to set job limits on your CUPS printers or users, these variables are easy to understand.

Sometimes, a user will try to send a print job but her program doesn't comply. A standard `Timeout` variable is set to close the CUPS connection; the default is 300 seconds.

```
#Timeout 300
```

Network Browsing

The browse parameters in CUPS relate to whether other computers on your network (or even other networks) can see the printers that you've configured with your CUPS server. By default, `Browsing` is on; other parameters determine how other computers see your CUPS printers.

```
#Browsing on
```

There are two protocols that you can configure for CUPS browsing: CUPS and SLPv2. CUPS broadcasts printer information; SLPv2 is the second version of the Service Location Protocol (SLP), which allows other computers to find available services.

Either protocol can be configured to collect and distribute information on shared printers on the network. The default is CUPS; if you want to use SLPv2, your network needs access to at least one SLPv2 directory agent. While CUPS is the default protocol, you can configure either or both with one of the following commands:

```
#BrowseProtocols cups
#BrowseProtocols slp
#BrowseProtocols all
```

When your CUPS server broadcasts data on your shared printers, it needs a broadcast address. This is usually the broadcast IP address for your network and is designated as `BrowseAddress`. If your network includes a dial-up connection, you can set `BrowseAddress` to `@LOCAL`; or, if you want browsing only on the network connected to your `eth2` network card, use `@IF(eth2)`. You can use as many `BrowseAddress` commands as you need. Here are some examples:

```
#BrowseAddress 192.168.99.255
#BrowseAddress 10.255.255.255
#BrowseAddress @IF(eth1)
```

If your printer names are self-explanatory (`hplaser@joescomp`, for example), you don't have to specify the full location of the printer. CUPS assumes you have some skill in this area, so the `BrowseShortNames` variable is set to `Yes`. If you're in a big organization with large numbers of printers, and you want extended data on each printer, set it to `No`.

```
#BrowseShortNames Yes
```

Whenever you add or share a new CUPS printer, CUPS needs to update the list of available printers. This is controlled through the `BrowseInterval` variable, which is set to 30 seconds by default.

```
#BrowseInterval 30
```

Alternatively, you could set `BrowseInterval` to 0, which means that information on new CUPS printers will not be sent automatically to other computers. However, you can configure another CUPS server to find your printer browse list. For example, the following command gets the list of printers from a CUPS server at 192.168.0.222 on port 631:

```
#BrowsePoll 192.168.0.222:631
```

Whatever you do, don't set `BrowseTimeout` to a value lower than `BrowseInterval`. If you do, printers are removed from your list before they're shared with the rest of the network. The default is 300 seconds.

```
#BrowseTimeout 300
```

If you want to provide access to other networks, use the `BrowseRelay` variable. The following are examples of commands you'd use to send the list of your shared CUPS printers to computers on other

networks. The first address or interface must be on the local network. If you're using IP addresses, the second address can be a broadcast address for the other network.

```
#BrowseRelay 192.168.0.222 10.12.15.255
#BrowseRelay 192.168.0.0/24 10.12.15.255
```

The default port for CUPS broadcasts is the standard TCP/IP port for the Internet Print Protocol (IPP), 631. You could make your system a bit more secure by specifying a different port, but you'd have to make sure all other computers on your network are looking for printers on that different port by using the `BrowsePort` variable:

```
#BrowsePort 631
```

Browse Security

You can limit the computers that are allowed to browse your list of CUPS printers. By default, `BrowseAllow` accepts data from all addresses, and `BrowseDeny` does not deny access to any computer. You can specify networks by their IP address, network address, or domain name in a number of ways. Here are examples of valid commands:

```
# BrowseAllow 10.12.0.0/24
# BrowseAllow 10.12.0.0/255.255.0.0
# BrowseAllow all
# BrowseDeny *.example.com
# BrowseDeny none
# BrowseDeny @IF(eth1)
```

But what comes first, Allow or Deny? That's determined by the `BrowseOrder` variable. If it's set to

```
#BrowseOrder allow,deny
```

computers are allowed to see your list of shared printers, unless specifically listed in a `BrowseDeny` command. Conversely, the following command allows access only if the computer is listed in a `BrowseAllow` command:

```
#BrowseOrder deny,allow
```

Note *Naturally, if you want to specify a domain or a hostname, you need to set* `HostNameLookups` *to* `On`*.*

System Security

The area of security is where `cupsd.conf` looks most like an Apache configuration file. While the default CUPS group is sys, as defined by the `SystemGroup` variable

```
#SystemGroup sys
```

you can configure `<Location />` containers to regulate access IP addresses, classes, jobs, encryption, and more. The standard Red Hat configuration allows access to the CUPS server only from the local computer.

```
<Location />
Order Deny,Allow
Deny from All
Allow From 127.0.0.1
</Location>
```

You can specify other IP addresses in regular or CIDR notation. If you have `HostNameLookups` set to `On` (not recommended), you can even use host or domain names. As shown here, you can limit access by class (the first example limits access to a class named `AnyPrinter`) or by printer (the second example limits access to a specific printer named HPLaserJet) to the 192.168.1.0 network address:

```
<Location /AnyPrinter>
Order Deny,Allow
Deny from All
Allow From 127.0.0.1
</Location>

<Location /AnyPrinter/HPLaserJet>
Order Deny,Allow
Deny from All
Allow From 192.168.1.0/24
</Location>
```

Other containers allow you to regulate administrative operations, as described in Table 20.11.

Table 20.11: Location Container Options

Container	Description
`<Location />`	Associated with all CUPS print operations.
`<Location /admin>`	Associated with CUPS administrative operations; it may be a good idea to limit administrative access to CUPS.
`<Location /classes>`	Associated with limits on all configured CUPS printer classes.
`<Location /classes/`*`classname`*`>`	Associated with limits on the CUPS printer class named *`classname`*.
`<Location /jobs>`	Associated with limits on print job management.
`<Location /printers>`	Associated with limits administrative access on managing all printers.
`<Location /printers/`*`printname`*`>`	Associated with limits administrative access on managing the printer named *`printname`*.

Don't forget to end your containers with the `</Location>` command. Besides `Order`, `Deny`, and `Allow`, you can add other commands to a `<Location />` container. They are described in Table 20.12.

TABLE 20.12: LOCATION DIRECTIVE COMMANDS/DESCRIPTIONS

COMMAND	DESCRIPTION
`Allow`	Used for computers or interfaces allowed to access the specified printer or class.
`Anonymous`	Indicates that no username or password is required; generally the default.
`AuthClass`	Specifies required authentication; options include `Anonymous`, `User`, `System`, and `Group`.
`AuthGroupName`	Sets the name of the group associated with a `Group AuthClass`.
`AuthType`	Defines the type of required usernames and passwords; options include `None`, `Basic` using `/etc/passwd`, `Digest` and `Basic Digest` using `/etc/cups/passwd.md5`.
`Deny`	Used for computers or interfaces not allowed to access the specified printer or class.
`Encryption`	Specifies whether encryption is required for usernames and passwords; options include `Never`, `IfRequested`, `Required`, and `Always`.
`Limit`	Limits allowed CUPS request commands.
`LimitExcept`	Specifies prohibited CUPS request commands.
`Order`	Specifies how CUPS reads the `Deny` and `Allow` commands.
`Require`	Limits access to a `group`, a `user`, or all users with `valid-user`.

PRINTER CLASSES

You don't have to configure a class for each CUPS printer. You can set up `ImplicitClasses` for different printers with the same name, such as HPLaserJet. Print jobs to an Implicit Class are sent to the printer with the first available queue. `ImplicitClasses` is on by default.

```
#ImplicitClasses On
```

You can set the Implicit Class name to `AnyPrinter` by setting `ImplicitAnyClasses` to `On`. It is off by default.

```
#ImplicitAnyClasses Off
```

If you're using `ImplicitClasses`, your users don't really need to know about individual printers in a class. If `ImplicitClasses` is on, the `HideImplicitMembers` variable is on by default.

```
#HideImplicitMembers On
```

Printer Management

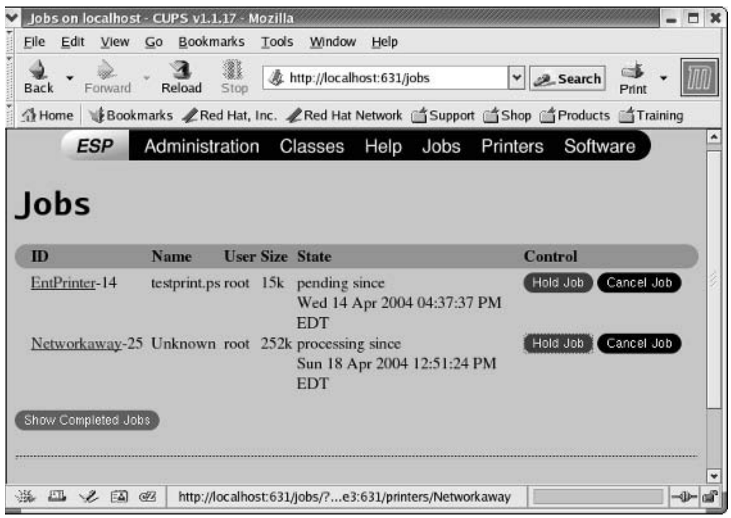
Once you've configured CUPS, you can use the CUPS GUI tool to manage current print jobs. You can also set up the `cups-lpd` service to allow you to use most standard LPD commands, including `lpr`,

`lpq`, and `lprm`. These commands are covered later in this chapter. Finally, you can monitor the CUPS log files in the `/var/log/cups` directory for status, errors, and suspicious access attempts.

Job Management

It's easy to manage active CUPS print jobs. The CUPS Jobs screen in Figure 20.16 shows two different print jobs. If you need to print job Networkaway-25 first, you click EntPrinter-14's Hold Job button. CUPS displays a message that "Job 14 has been held from printing," and the other job starts automatically.

Figure 20.16
Managing CUPS print jobs



Job EntPrinter-14 is held in the print queue until you return to the Jobs menu and click the Release Job button.

Activating LPD Commands

To activate LPD-style commands for a CUPS server, you need to activate the `cups-lpd` service in the `/etc/xinetd.d` directory. You can activate this service with the `chkconfig cups-lpd on` command. More information on managing `xinetd` services is available in Chapter 18.

You may need to activate `cups-lpd` for some applications that were originally designed for an LPD-style interface.

CUPS Log Files

CUPS log files, briefly described earlier in this chapter, are normally stored in the `/var/log/cups` directory. The `access_log` file lists the computer along with the date and time of access to the CUPS server. The example shown in Figure 20.17 lists access from the default local computer (localhost) as well as two other computers on my network.

FIGURE 20.17
CUPS *access_log* file

```
192.168.1.21 - - [18/Apr/2004:14:55:33 -0400] "GET /jobs HTTP/1.1" 200 0
localhost - - [18/Apr/2004:14:55:33 -0400] "POST / HTTP/1.1" 200 220
localhost - - [18/Apr/2004:14:55:33 -0400] "POST / HTTP/1.1" 200 122
192.168.1.21 - - [18/Apr/2004:14:55:33 -0400] "GET /jobs HTTP/1.1" 200 1613
192.168.1.21 - - [18/Apr/2004:14:55:36 -0400] "GET /images/show-completed.gif HT
TP/1.1" 200 337
localhost - - [18/Apr/2004:14:55:38 -0400] "POST / HTTP/1.1" 200 220
localhost - - [18/Apr/2004:14:55:43 -0400] "POST / HTTP/1.1" 200 220
localhost - - [18/Apr/2004:14:55:48 -0400] "POST / HTTP/1.1" 200 220
localhost - - [18/Apr/2004:14:55:53 -0400] "POST / HTTP/1.1" 200 220
localhost - - [18/Apr/2004:14:55:58 -0400] "POST / HTTP/1.1" 200 220
192.168.1.13 - - [18/Apr/2004:14:56:01 -0400] "GET / HTTP/1.1" 200 1599
192.168.1.13 - - [18/Apr/2004:14:56:01 -0400] "GET /cups.css HTTP/1.1" 200 87
192.168.1.13 - - [18/Apr/2004:14:56:01 -0400] "GET /images/navbar.gif HTTP/1.1"
200 2869
192.168.1.13 - - [18/Apr/2004:14:56:03 -0400] "GET /jobs HTTP/1.1" 200 0
localhost - - [18/Apr/2004:14:56:03 -0400] "POST / HTTP/1.1" 200 220
localhost - - [18/Apr/2004:14:56:04 -0400] "POST / HTTP/1.1" 200 112
192.168.1.13 - - [18/Apr/2004:14:56:03 -0400] "GET /jobs HTTP/1.1" 200 1613
192.168.1.13 - - [18/Apr/2004:14:56:04 -0400] "GET /images/show-completed.gif HT
TP/1.1" 200 337
localhost - - [18/Apr/2004:14:56:08 -0400] "POST / HTTP/1.1" 200 220
~
                                                          3574,11       Bot
```

The `error_log` file lists more than just standard errors; as shown in Figure 20.18, it also lists basic activity of the CUPS server, and it even identifies a network problem.

FIGURE 20.18
CUPS *error_log* lists more than just errors.

```
I [18/Apr/2004:04:03:55 -0400] Sending browsing info to ffffffff:631
I [18/Apr/2004:04:03:55 -0400] Listening to 0:631
I [18/Apr/2004:04:03:55 -0400] Configured for up to 100 clients.
I [18/Apr/2004:04:03:55 -0400] Allowing up to 10 client connections per host.
I [18/Apr/2004:04:03:55 -0400] LoadPPDs: Read "/etc/cups/ppds.dat", 2771 PPDs...
I [18/Apr/2004:04:03:56 -0400] LoadPPDs: No new or changed PPDs...
I [18/Apr/2004:09:13:57 -0400] Job 35 queued on 'CosmicC' by 'root'.
I [18/Apr/2004:09:13:57 -0400] Started backend /usr/lib/cups/backend/parallel (P
ID 30634) for job 35.
I [18/Apr/2004:09:14:41 -0400] Job 36 queued on 'CosmicC' by 'root'.
I [18/Apr/2004:09:14:41 -0400] Started backend /usr/lib/cups/backend/parallel (P
ID 30637) for job 36.
E [18/Apr/2004:12:56:22 -0400] SendBrowseList: sendto failed for browser 1 - Net
work is unreachable.
I [18/Apr/2004:14:55:26 -0400] Started "/usr/lib/cups/cgi-bin/classes.cgi" (pid=
1838)
I [18/Apr/2004:14:55:33 -0400] Started "/usr/lib/cups/cgi-bin/jobs.cgi" (pid=186
1)
I [18/Apr/2004:14:56:03 -0400] Started "/usr/lib/cups/cgi-bin/jobs.cgi" (pid=186
3)
~
~
~
"/var/log/cups/error_log" 14L, 1105C                      14,16         All
```

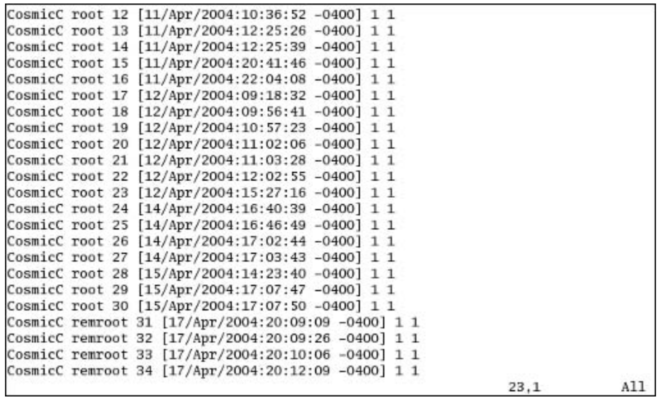
Finally, the `page_log` file lists any job that's been sent to the queue, even if it was cancelled. An example of this file is shown in Figure 20.19. Note the `remroot` jobs, which were sent from other computers on this network.

Printer Management Commands

If you are using CUPS and have activated the `cups-lpd` service in `/etc/xinetd.d`, you can still use several different LPD commands. While the output is not identical, the results should be sufficient for Linux printer administrators.

Four basic commands are associated with the LPD: the Line Printer Request, `lpr`; the Line Printer Query, `lpq`; the Line Printer Remove, `lprm`; and the Line Printer Control, `lpc`. These are sometimes known as the *lp commands*.

FIGURE 20.19
CUPS *page_log* lists print jobs.



```
CosmicC root 12 [11/Apr/2004:10:36:52 -0400] 1 1
CosmicC root 13 [11/Apr/2004:12:25:26 -0400] 1 1
CosmicC root 14 [11/Apr/2004:12:25:39 -0400] 1 1
CosmicC root 15 [11/Apr/2004:20:41:46 -0400] 1 1
CosmicC root 16 [11/Apr/2004:22:04:08 -0400] 1 1
CosmicC root 17 [12/Apr/2004:09:18:32 -0400] 1 1
CosmicC root 18 [12/Apr/2004:09:56:41 -0400] 1 1
CosmicC root 19 [12/Apr/2004:10:57:23 -0400] 1 1
CosmicC root 20 [12/Apr/2004:11:02:06 -0400] 1 1
CosmicC root 21 [12/Apr/2004:11:03:28 -0400] 1 1
CosmicC root 22 [12/Apr/2004:12:02:55 -0400] 1 1
CosmicC root 23 [12/Apr/2004:15:27:16 -0400] 1 1
CosmicC root 24 [14/Apr/2004:16:40:39 -0400] 1 1
CosmicC root 25 [14/Apr/2004:16:46:49 -0400] 1 1
CosmicC root 26 [14/Apr/2004:17:02:44 -0400] 1 1
CosmicC root 27 [14/Apr/2004:17:03:43 -0400] 1 1
CosmicC root 28 [15/Apr/2004:14:23:40 -0400] 1 1
CosmicC root 29 [15/Apr/2004:17:07:47 -0400] 1 1
CosmicC root 30 [15/Apr/2004:17:07:50 -0400] 1 1
CosmicC remroot 31 [17/Apr/2004:20:09:09 -0400] 1 1
CosmicC remroot 32 [17/Apr/2004:20:09:26 -0400] 1 1
CosmicC remroot 33 [17/Apr/2004:20:10:06 -0400] 1 1
CosmicC remroot 34 [17/Apr/2004:20:12:09 -0400] 1 1
                                                   23,1          All
```

LPR

When you have Linux read the contents of a file with the `cat` command, the shell sends the result to standard output, which normally means you see the result on your screen. In contrast, when you use `lpr`, the shell sends the result to a spool file on the local computer, then on to a print server computer, and finally to the printer. The `lpr` command is effectively a client. When it produces a spool file, the result is processed by the `lpd` server on a local or remote network.

Therefore, when you run a command such as `lpr` *`file`*, the shell sends the result to the default printer as configured in `/etc/printcap`. Alternatively, you can send the print job to a different printer. For example, if colorprinter is configured in `/etc/printcap`, the following command sends the job to that printer:

```
# lpr -Pcolorprinter file1
```

NOTE *When using the* `lpr` *command to specify a printer, there's no space between the* `-P` *and the name of the printer.*

Other variations on the `lpr` command are shown in Table 25.13.

TABLE 20.13: ***LPR*** **COMMANDS**

COMMAND	**RESULT**
`lpr -h` *`file1`*	Prints *`file1`* without a job control page, which normally contains the user account and hostname of the source computer. The job control page is also known as the *burst* page.
`lpr -P`*`other file1`*	Prints *`file1`* to the printer named *other*, as defined in the `/etc/printcap` file.
`lpr -s` *`file1`*	Creates a symbolic link to *`file1`*, which avoids creating a spool file. This was required for larger (>1MB) files on the Berkeley Standard Distribution version of `lpr`. Red Hat Linux 9 uses the LPRng program, which makes this unnecessary.

LPQ

The `lpq` command gives you the current print queue. There are three basic options, as shown in Table 20.14. This command also includes a list of job numbers, which you may need for the `lprm` command.

Table 20.14: *LPQ* **Command Examples**

Command	Result
`lpq`	Returns the current print queue for the default printer, as defined in your `/etc/printcap` file.
`lpq -P printer`	Returns the print queue for the named *`printer`*. Uses the name as defined in your `/etc/printcap` file.

LPRM

If a print job isn't already in your printer's memory, the `lprm` command can delete print jobs currently in your queue. With `lprm`, you can remove a print job in one of three ways: by print job number, by user, or by printer. Table 20.15 shows examples of this command.

Table 20.15: *LPRM* **Command Examples**

Command	Result
`lprm 188`	Removes print job 188, as defined in the output to the `lpq` command
`lprm -P hp2 mj`	Removes print jobs of user mj from the printer labeled *hp2* in your `/etc/printcap` file

LPC

The `lpc` command allows you to control a number of characteristics of each printer. As shown in Table 20.16, this command lets you check printer status, kill active print jobs, or even redirect jobs to a different printer.

Table 20.16: *LPC* **Command Examples**

Command	Result
`lpc -P canon1 status`	Displays the status of the printer named *canon1*. In other words, the output tells you whether you can send print jobs to a queue, the number of jobs in the queue, whether the printer will accept jobs, and communication status with the printer.
`lpc disable`	Disables sending jobs (*spooling*) to a print queue for the default printer. Opposite of `lpc enable`.
`lpc start`	Restarts transfers from the print queue.
`lpc stop`	Stops communication between the print queue and your printer.

Summary

In this chapter, we examined the two major options for print servers, CUPS and LPD. CUPS is the new default Red Hat Enterprise Linux print server. LPD, which has been the default for years, will be removed in a future release of Red Hat Enterprise Linux, so it is important for you to learn about CUPS.

CUPS is short for the Common Unix Printing System. It provides a common way for Linux and other Unix-type operating systems to work with the Internet Printing Protocol (IPP). IPP is becoming the standard print server for a wide variety of operating systems, so it makes sense to move to CUPS.

Red Hat includes a GUI tool to help configure CUPS configuration files, which you can start with the `redhat-config-printer` command. This can simplify the printer configuration process, as the language associated with the configuration files in `/etc/cups` may be confusing for newer administrators.

CUPS includes a graphical browser-based tool available on port 631. With the CUPS tool, you can configure printers, classes of different printers, and print jobs. However, print drivers for the browser-based tool are limited. On the other hand, the browser-based tool can help you configure a group of printers as a class; print jobs are automatically sent to the first available printer in that class.

CUPS configuration files are stored in the `/etc/cups` directory. While the main CUPS configuration file, `cupsd.conf`, is long, it is based on the same format as Apache configuration files. A substantial number of settings are available for everything from job size to logs to security.

Once you've configured printers and print classes, it's easy to use the CUPS web-based configuration tool to manage printers and print jobs. For example, in the Jobs section, you can hold a print job to allow a higher priority job through. You can also check the status of current printers with the `lpstat` and `lpadmin` commands. If you activate the `cups-lpd` daemon, you can also use several basic LPD-style commands. For example, you can manage the printers and queues with several basic commands, including `lpr`, `lpq`, `lprm`, and `lpc`.

In Chapter 21, we'll look at mail servers and clients, with a focus on configuring sendmail for your network.

Chapter 21

Mail Services

In this chapter, we'll look at one of the essential applications for any computer that is connected to a network: e-mail. Various server services are used to send and to receive e-mail, and each server is associated with one or more protocols. While e-mail clients are relatively straightforward to configure, e-mail servers have a number of rich and complex options.

There are several basic TCP/IP protocols related to e-mail. The two most common protocols for receiving e-mail are the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP). The Simple Mail Transfer Protocol (SMTP) is an important protocol for sending mail from your network.

The most common SMTP e-mail server on the Internet is *sendmail*. While the basic sendmail configuration file is complex, Red Hat Enterprise Linux includes a macro file that is easy to customize based on what you need. You can edit this file and then use a macro processor to generate a custom configuration file for sendmail. This configuration file also helps you address security needs by setting up responses for FQDN that you can't verify and domains where you don't want to send mail.

Postfix is an alternative to sendmail that is probably already installed on your Red Hat Enterprise Linux system. It is the successor to the VMailer and IBM Secure Mailer systems. Although you can't run both services at once, you can use `redhat-switch-mail` to switch between these servers.

There are two basic servers for incoming e-mail, based on the IMAP4 and POP3 protocols. You can create your own incoming e-mail server, or you can set up your e-mail clients to use incoming e-mail servers from an outside e-mail provider. Red Hat Enterprise Linux includes both of these servers, in regular and secure versions, in one RPM package.

Most computer users are familiar with at least one e-mail client. The principles behind them are the same. They take the e-mail data and format it in a fashion to which you can easily read and reply. Linux includes both text and graphical e-mail clients. This chapter covers the following topics:

- Examining general mail services
- Configuring sendmail
- Setting up Postfix
- Using incoming e-mail servers
- Configuring mail clients

Examining General Mail Services

Three kinds of mail services are available: Message Transfer Agents (MTA), Mail Delivery Agents (MDA), and Mail User Agents (MUA).

An MTA is a server that sends e-mail through a network. Linux uses MTA agents such as sendmail, which uses the SMTP protocol to send e-mail over a TCP/IP network such as the Internet.

An MDA is a mail processor. It takes messages from the Internet and stores them in servers or spools where mail readers (a.k.a. MUAs)—such as `mutt`, Mozilla Mail, KMail, and Evolution—can read them. The most common example of an MDA is procmail. While the `procmail-*` RPM is installed by default, it works seamlessly with a properly configured sendmail (and other outgoing e-mail server) package.

An MUA is an application that helps you send and receive e-mail through these servers. Most users are familiar with at least one MUA, such as those listed, or Lotus Notes, Netscape, or `pine`. When you prepare and send an e-mail message, you're using an MUA to send a message to an MTA such as sendmail.

The two major protocols for receiving e-mail are POP3 and IMAP4. Mail servers configured to either protocol are simply one step in the MDA process.

NOTE *All you need to do to configure POP3 or IMAP4 is enable their respective configuration files (*`pop3s` *and* `imaps`*) in the* `/etc/xinted.d` *directory. See Chapter 18 for more information.*

Key Protocols

A substantial number of TCP/IP protocols are involved in sending an e-mail message from one user to another. We've mentioned three of them: SMTP, POP3, and IMAP4.

As a Linux administrator, you've probably set up an e-mail server at some point in time. While sendmail is the most important of the SMTP servers, several alternatives are available, including Exim, Postfix, and Qmail.

As a Linux administrator, you may also help users configure their e-mail clients. Generally, you'll need to know the names of any incoming e-mail servers on your network or with your ISP. This may include the name of the mail exchanger (`MX`) record that you created in your DNS server in Chapter 19. But this chapter is focused on outgoing mail.

Older mail servers used the Unix-to-Unix Copy Protocol (UUCP), which sent messages directly from computer to computer. If the message had to go to a different network, you would have to specify each computer on the path. Needless to say, this has become unwieldy with the expansion of the Internet.

Alternate Mail Servers

While the rest of this chapter is focused on sendmail, there are alternatives, based on the search for the easy-to-configure e-mail server. Packages for each of these systems (except the commercial version of sendmail, which is Sendmail) are available from sources such as `www.rpmfind.net`.

Commercial Sendmail Unlike the free version included with Red Hat Enterprise Linux, the commercial version of Sendmail is designed for the enterprise. In other words, it can help you serve

many thousands of users. It is even configurable for mobile clients. More information is available at `www.sendmail.com`.

Exim The Exim MTA was developed at Cambridge (U.K.) and is licensed under the GPL. While based on an older MTA known as Smail, it can also help you verify user addresses and refuse e-mail. This helps you minimize spam sent to users on your system. More information is available at `www.exim.org`.

Qmail The Qmail MTA is another alternative to sendmail. According to `www.qmail.org`, Qmail is used by an impressive list of Internet sites. The developer, D. J. Bernstein, offered a cash reward in 1997 for the first person to find a security hole in this system (`cr.yp.to/qmail/guarantee.html`). His offer still stands.

Smail The Smail MTA is reportedly easier to configure than sendmail. It also includes support for blocking messages. In addition, it helps you protect yourself from "spoofed" messages that try to mask themselves as coming from trusted sites. While no official website exists for this MTA, the developers can be found at `www.planix.com`.

Switching Between Mail Services

If you install the Mail Server package group, you get both sendmail and Postfix by default. You can't run both services at the same time. You can disable one and enable the other with the `service` and `chkconfig` commands, or you can use the Red Hat Mail System Switcher, also known as `redhat-switch-mail`.

As of this writing, you can start this utility only from a GUI command-line interface; there is no corresponding entry in the GNOME Main Menu. Figure 21.1 displays the resulting `redhat-switch-mail` window. Changes are made to the desired mail daemons at the appropriate runlevels.

Configuring sendmail

As with most complex Linux services, sendmail components can be installed from a number of RPM packages. There are many key configuration files, over and above the `sendmail.cf` configuration file and `sendmail.mc` macro.

FIGURE 21.1
Switching mail systems

With the latest version of sendmail, the configuration files are now split into two parts. When sendmail receives e-mail, it uses `sendmail.cf`. When sendmail sends e-mail, it uses `submit.cf`.

Once you get sendmail up and running, you can modify various configuration files to promote security.

NOTE *This is far from a comprehensive discussion on sendmail; there are 1000-page books available just on this service. One good reference is* Linux Sendmail Administration, *by Craig Hunt (Sybex, 2001).*

Packages

The only RPMs you need for a working sendmail configuration are the two `sendmail-*` RPMs, whose packages are installed as part of the Mail Server package group. You can install the group using the Red Hat Package Management tool described in Chapter 10. Alternatively, you can just install the RPMs. The Red Hat Enterprise Linux sendmail packages are listed in Table 21.1; as you might remember from Chapter 10, you can use the `rpm -q` *`packagename`* command to see if they're installed. Once they're installed, you can use the `rpm -ql` *`packagename`* command to see the associated files.

TABLE 21.1: SENDMAIL RPM PACKAGES

PACKAGE	FUNCTION
`sendmail-*`	The sendmail MTA software
`sendmail-cf-*`	Tools and templates for creating a wide variety of sendmail configuration files

Basic Configuration Files

There is more to sendmail than just the basic configuration file, `sendmail.cf`, and the macro file, `sendmail.mc`. There are other configuration files in the `/etc/mail` directory. As with many other daemons, sendmail has a control file in `/etc/sysconfig`. You can set it to forward e-mail to a different user through `/etc/aliases`.

BASIC */ETC/SYSCONFIG/SENDMAIL*

The `/etc/sysconfig/sendmail` file is fairly simple.

```
DAEMON=yes
QUEUE=1h
```

The `DAEMON=yes` entry sets sendmail to listen for messages on TCP/IP port 25, which is associated with SMTP. The `QUEUE=1h` entry tells sendmail to try to deliver queued mail every hour.

SENDMAIL ALIASES

The `/etc/aliases` file is also simple. It specifies the users who should really receive e-mail. For example, if you try to send mail to a service such as `ftp@localhost`, the following entry redirects that mail to `root@localhost`:

```
ftp:     root
```

Or, you can redirect e-mail from a former to a current employee.

```
byeltsin:    vputin
```

SENDMAIL */ETC/MAIL* CONFIGURATION FILES

There are a number of files in `/etc/mail` that you can use to configure sendmail or to set up databases to regulate how sendmail works. If you want to enable these configuration files, you generally need an entry in the `sendmail.mc` macro file. If there is a `.db` file, you can in most cases convert a text file such as `access` to `access.db` by using the `makemap` command (which is also run when you process the configuration files in this directory with the `make -C /etc/mail` command).

access* and *access.db Configures domains or e-mail addresses; e-mail from these sources can be dropped (`DISCARD`), rejected with an error message (`REJECT`), or sent to the specified address (`RELAY`). You can minimize unwanted e-mail by dropping or rejecting it from specific domains or e-mail addresses. Look at the `/etc/mail/access` file for examples.

domaintable* and *domaintable.db Maps two different domains. These files are useful if you've converted your domain name and others are still sending e-mail to your users' old e-mail addresses. If you've just converted your domain name from `dictatorsrus.com` to `democracyisus.com`, you could add the following line to your `domaintable` file:

```
dictatorsrus.com    democracyisus.com
```

helpfile Provides help for commands available at the sendmail prompt. You can get to the sendmail prompt with the `telnet localhost 25` command.

local-host-names Contains aliases or other hostnames for your sendmail server. Just enter other names for your sendmail server computer on individual lines in this file.

mailertable* and *mailertable.db Lets you specify an unusual e-mail server type for a specific address; rarely used.

Makefile Lets you compile different options; it allows the `make -C /etc/mail` command to process all files in the `/etc/mail` directory.

sendmail.cf* and *sendmail.mc Allows you to configure sendmail. `sendmail.cf` is the configuration file; `sendmail.mc` is a macro file that can be processed into the configuration file. More information on these files is available later in this chapter.

spamassassin Supports configuration of the SpamAssassin spam reducer (`www.spamassassin.org`). You can enable SpamAssassin for all users by adding the following line to the `/etc/procmailrc` configuration file (which you may need to create):

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

statistics Contains statistics for sendmail usage. Run the `mailstats` command to read this file.

submit.cf* and *submit.mc Allows you to limit sendmail usage to specific groups. The syntax in the default `submit.mc` file is the same as in `sendmail.mc`. More information on `submit.mc` is available later in this chapter.

trusted-users Lets you list users who can send e-mail on behalf of your other users. Rarely used; would you ever want to give anyone this kind of power?

virtualusertable* and *virtualusertable.db Supports e-mail forwarding; similar to the `/etc/aliases` file, for external users.

Understanding *sendmail.mc*

The `/etc/mail/sendmail.cf` configuration file can be intimidating—it is on the order of 2,000 lines long! By comparison, the `/etc/mail/sendmail.mc` file, at about 140 lines, is easy to read and understand. Once you've configured this file to your liking, you can use an appropriate `make` command or the `m4` macro processor to generate the custom `sendmail.cf` file you need. Take a look at this file; I've included additional comments where appropriate. As you probably won't need to modify most of this file, my comments are limited. As sendmail is a complex topic, please refer to *Linux Sendmail Administration* by Craig Hunt (Sybex, 2001) for more information.

NOTE *The quote marks inside the parentheses in* `sendmail.mc` *may not be what you expect: They start with a back quote (`) and end with a single quote (') mark. The back quote is the character above the Tab key on a U.S. keyboard.*

The `divert(-1)` command is a standard way to start the `sendmail.mc` file; if paired with `divert(0)`, all lines between these commands are ignored as comments.

```
divert(-1)dnl
```

All lines that start with `dnl` are comments; these particular comments include one way to process the `sendmail.mc` file. Alternatively, you can still regenerate `/etc/mail/sendmail.cf` with the `m4 sendmail.mc > sendmail.cf` command.

```
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package is
dnl # installed and then performing a
dnl #
dnl #     make -C /etc/mail
dnl #
```

The following `include` command adds the `cf.m4` command as a macro processing prototype; by default, it requires installation of the `sendmail-cf-*` RPM.

```
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
```

The `VERSIONID` is the label associated with each sendmail configuration file.

```
VERSIONID(`setup for Red Hat Linux ')dnl
```

Naturally, any `OSTYPE` command specifies the operating system, in this case, `linux`.

```
OSTYPE(`linux')dnl
```

The `define` command, as follows, coordinates your sendmail server with an outgoing e-mail server, presumably outside your network. If you want to activate this command, delete the `dnl` in front of `define` and replace `smtp.your.provider` with the outgoing (SMTP) e-mail server address of your ISP.

```
dnl #
dnl # Uncomment and edit the following line if your outgoing mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(`SMART_HOST',`smtp.your.provider')
dnl #
```

Generally, no changes are required to the following commands; see *Linux Sendmail Administration* for more information:

```
define(`confDEF_USER_ID',``8:12'')dnl
dnl define(`confAUTO_REBUILD')dnl
define(`confTO_CONNECT', `1m')dnl
define(`confTRY_NULL_MX_LIST',true)dnl
define(`confDONT_PROBE_INTERFACES',true)dnl
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
dnl define(`STATUS_FILE', `/etc/mail/statistics')dnl
define(`UUCP_MAILER_MAX', `2000000')dnl
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')dnl
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,restrictqrun')dnl
```

The two following commands that start with `` define(`confAUTH_OPTIONS' `` are mutually exclusive. TLS is Transport Layer Security, which is the successor to SSL, the Secure Socket Layer.

```
define(`confAUTH_OPTIONS', `A')dnl
dnl #
dnl # The following allows relaying if the user authenticates, and disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(`confAUTH_OPTIONS', `A p')dnl
```

If you need to prevent plain-text logins to your sendmail server, change these two commands so they read as follows:

```
dnl define(`confAUTH_OPTIONS', `A')dnl
define(`confAUTH_OPTIONS', `A p')dnl
```

Now let's continue with the default `sendmail.mc` file. As defined by the comments, the following two commands relate to authorization methods:

```
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
```

```
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl #
dnl TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN
➥ PLAIN')dnl
```

The following commands allow you to use any SSL certificates on your system with sendmail. For more information on SSL certificates, see Chapter 25. The certificates you can create in that chapter for Apache can also apply here.

```
dnl #
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl #     make -C /usr/share/ssl/certs usage
dnl #
dnl define(`confCACERT_PATH',`/usr/share/ssl/certs')
dnl define(`confCACERT',`/usr/share/ssl/certs/ca-bundle.crt')
dnl define(`confSERVER_CERT',`/usr/share/ssl/certs/sendmail.pem')
dnl define(`confSERVER_KEY', `/usr/share/ssl/certs/sendmail.pem')
dnl #
```

The following `define` command supports integration with the Lightweight Directory Access Protocol (LDAP), which provides detailed user information and can therefore replace the following `/etc/aliases` and `/etc/mail/virtusertable.db` files. Integration of sendmail and LDAP is a complex topic beyond the scope of this book.

```
dnl # This allows sendmail to use a keyfile that is shared with OpenLDAP's
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(`confDONT_BLAME_SENDMAIL',`groupreadablekeyfile')dnl
dnl #
```

The following commands specify actions associated with e-mail that can't find the destination. For example, if your e-mail has trouble getting to the recipient's e-mail server, you get a warning message after four (4) hours (`confTO_QUEUEWARN`) and an undeliverable message after five (5) days (`confTO_QUEUERETURN`).

```
dnl define(`confTO_QUEUEWARN', `4h')dnl
dnl define(`confTO_QUEUERETURN', `5d')dnl
dnl define(`confQUEUE_LA', `12')dnl
dnl define(`confREFUSE_LA', `18')dnl
define(`confTO_IDENT', `0')dnl
dnl FEATURE(delay_checks)dnl
FEATURE(`no_default_msa',`dnl')dnl
```

This `FEATURE` command sets the default sendmail shell, `smrsh`. The `mailertable.db` associates different domain names.

```
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
```

```
FEATURE(`mailertable',`hash -o/etc/mail/mailertable.db')dnl
FEATURE(`virtusertable',`hash -o/etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The -t option will retry delivery if e.g. the user runs over his quota.
dnl #
FEATURE(local_procmail,`',`procmail -t -Y -a $h -d $u')dnl
FEATURE(`access_db',`hash -T<TMPF> -o/etc/mail/access.db')dnl
FEATURE(`blacklist_recipients')dnl
```

If the root user tries to log in, the `EXPOSED_USER` command requires the full e-mail address.

```
EXPOSED_USER(`root')dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
```

By default, sendmail listens for and processes e-mail only from the local computer. If you want this sendmail server to work for other computers on your network, add a `dnl` in front of this command and remove it from one of the other `DAEMON_OPTIONS` commands that follow:

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
dnl # mail from MUAs that authenticate. Roaming users who can't reach their
dnl # preferred sendmail daemon due to port 25 being blocked or redirected find
dnl # this useful.
dnl #
```

If you activate the following `DAEMON_OPTIONS` command, sendmail will listen for e-mail from users who send their accounts and passwords—that is, whose e-mail mangers authenticate. This process works through TCP/IP port 587.

```
dnl DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465, but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587 followed
dnl # by STARTTLS is preferred, but roaming clients using Outlook Express can't
dnl # do STARTTLS on ports other than 25. Mozilla Mail can ONLY use STARTTLS
dnl # and doesn't support the deprecated smtps; Evolution <1.1.1 uses smtps
dnl # when SSL is enabled-- STARTTLS support is available in version 1.1.1.
dnl #
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
```

If you want to require secure connections to your sendmail server, you could activate this command, which requires the use of TLS. However, as noted in the comments (shown previously), you should not activate this command if your users work with Microsoft Outlook Express or Evolution below version 1.1.1.

```
dnl DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
dnl #
dnl # The following causes sendmail to additionally listen on the IPv6 loopback
dnl # device. Remove the loopback address restriction listen to the network.
dnl #
dnl # NOTE: binding both IPv4 and IPv6 daemon to the same port requires
/dnl #       a kernel patch
dnl #
```

Activate the following command if you've configured your network to use IPv6, as described in Chapter 15. This is the IPv6 equivalent of the default command noted earlier that accepts e-mail only from the local computer.

```
dnl DAEMON_OPTIONS(`port=smtp,Addr=::1, Name=MTA-v6, Family=inet6')dnl
dnl #
dnl # We strongly recommend not accepting unresolvable domains if you want to
dnl # protect yourself from spam. However, the laptop and users on computers
dnl # that do not have 24x7 DNS do need this.
dnl #
```

This `FEATURE` command means that sendmail doesn't do a reverse DNS lookup on an e-mail message. Unless you have reliable access to a DNS server and can accept the extra traffic, keep the command as is.

```
FEATURE(`accept_unresolvable_domains')dnl
dnl #
```

This `FEATURE` command allows the use of the `MX` records for a mail server as specified in a DNS database. See Chapter 19 for more information on DNS.

```
dnl FEATURE(`relay_based_on_MX')dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
```

The `LOCAL_DOMAIN` command specifies an alias for the local computer; `localhost.localdomain` is a default alias in `/etc/hosts`.

```
LOCAL_DOMAIN(`localhost.localdomain')dnl
dnl #
dnl # The following example makes mail from this host andany additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
```

This MASQUERADE_AS command changes the label that sendmail attaches to your outgoing e-mail. If you activate this command, change mydomain.com to the label you desire, typically used to specify e-mail from a subdomain. For example, if I'm on the mommabears.com network, I could set MASQUERADE_AS to linux.mommabears.com.

```
dnl MASQUERADE_AS(`mydomain.com')dnl
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
dnl FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
```

With the MASQUERADE_DOMAIN command, you can tell sendmail to handle e-mail addresses from other domains in the same way. For example, these commands, if active, set e-mail from these subdomains (localhost, localhost.localdomain, mydomainalias.com, and mydomain.lan) to the domain specified earlier with the MASQUERADE_AS command. Naturally, you'll want to substitute the name of your domain for mydomainalias.com and mydomain.lan.

```
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
```

The following MAILER commands specify the type of server that actually sends out the e-mail.

```
MAILER(smtp)dnl
MAILER(procmail)dnl
```

Revising *sendmail.mc*

Before you start, it's a good idea to make backups of your sendmail.cf and sendmail.mc files in your /etc/mail directory. I start with a complete backup of the /etc/mail directory to my home directory with the cp -ar /etc/mail ~ command.

To make this service work for your network, there are a couple of lines that you should change in the default sendmail.mc configuration file. First, this line limits the sendmail server to sending e-mail only to the specified address; 127.0.0.1 is the loopback address for the local computer:

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
```

Next, if you want to enable sendmail for your network, you need to disable this command by adding a dnl in front.

```
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
```

If you have reliable DNS access and high-speed Internet access, comment out this next line. It keeps sendmail from checking the domain associated with incoming e-mail addresses. You can comment out the line by putting `dnl` in front; when you restart the `sendmail` service, sendmail automatically starts checking domains.

```
FEATURE(`accept_unresolvable_domains')dnl
```

Understanding and Revising *submit.mc*

The `submit.mc` file is the macro file used to create `submit.cf`, the sendmail configuration file for outgoing e-mail. It is processed in the same way as `sendmail.mc`; fortunately, this file is simpler. Generally, you don't need to make any changes to this file—but it's helpful to understand this file to know what other files to configure.

These first commands are essentially the same as the first commands in `sendmail.mc` and are explained in that section of this chapter.

```
divert(-1)dnl
divert(0)dnl
include(`/usr/share/sendmail-cf/m4/cf.m4')
VERSIONID(`linux setup for Red Hat Linux')dnl
```

The `confCF_Version` command simply adds to the version name.

```
define(`confCF_VERSION', `Submit')dnl
```

This adds an operating system type, similar to the `OSTYPE(`linux')dnl` command in `sendmail.mc`.

```
define(`__OSTYPE__',`')dnl dirty hack to keep proto.m4 from complaining
```

`DECNET` is a network type common on older mainframe and microcomputers.

```
define(`_USE_DECNET_SYNTAX_', `1')dnl support DECnet
```

The `confTIME_ZONE` variable adds a time stamp.

```
define(`confTIME_ZONE', `USE_TZ')dnl
```

This setting avoids looking through any NIS list for users and passwords; the alternative source of usernames and passwords is `/etc/passwd`. If you've set up NIS for your network (see Chapter 23), you can comment out this line by adding a `dnl` in front.

```
define(`confDONT_INIT_GROUPS', `True')dnl
```

This sets the location of the process identifier (PID) file.

```
define(`confPID_FILE', `/var/run/sm-client.pid')dnl
```

The `confDIRECT_SUBMISSION_MODIFIERS` variable assumes standard "canonical" host names.

```
dnl define(`confDIRECT_SUBMISSION_MODIFIERS',`C')
```

The `use_ct_file FEATURE` reads `/etc/mail/trusted-users` for standard users.

```
FEATURE(`use_ct_file')dnl
dnl
dnl If you use IPv6 only, change [127.0.0.1] to [IPv6:::1]
```

This notes the message submission program (`msp`) on the local computer (127.0.0.1).

```
FEATURE(`msp', `[127.0.0.1]')dnl
```

In most cases, you don't need to change anything in this file; if you do, please remember to back it up first!

Processing and Reactivating sendmail

If you haven't already done so, now is a good time to back up your current `sendmail.cf` configuration file. Once you've made the desired changes, you'll want to use the `make -C /etc/mail` command to create new `sendmail.cf` and `submit.mc` configuration files. Then, restart the `sendmail` daemon with the following commands.

```
# make -C /etc/mail
# service sendmail restart
```

These commands won't work unless you've installed the `sendmail-cf-*` RPM. Naturally, you'll want to make sure that `sendmail` starts the next time you start Linux at appropriate runlevels with a command such as this:

```
# chkconfig --level 35 sendmail on
```

Setting Up Postfix

When you install the Red Hat Enterprise Linux Mail Server package group, you get a selection of two different mail servers. The sendmail server is more popular and well documented. However, I find that Postfix is somewhat easier to configure. The configuration files are stored in `/etc/postfix` and are similar to those for sendmail.

To configure Postfix, you can edit just a few settings in the `main.cf` configuration file. As it is easier to navigate, there is no corresponding macro file. The service is contained in the `postfix` RPM.

NOTE *This is far from a comprehensive discussion on Postfix; it just includes the elements required to get this service working on Red Hat Enterprise Linux 3. For more information, see* `www.postfix.org`.

Basic Files and Packages

The only RPM you need for a working Postfix configuration is the `postfix-*` RPM, which is installed by default as part of the Mail Server package group. You can install the group using the Red Hat Package Management tool described in Chapter 10. Alternatively, you can just install the RPM.

There is more to Postfix than just the basic configuration file, `main.cf`. There are a number of configuration files in the `/etc/postfix` directory, as described here. You may note that many of these

files help you work with changes in users, e-mail addresses, and domains. Generally, all you'll need to edit before using Postfix for the first time is `main.cf`, and possibly `access` and `aliases`.

aliases User substitutes; most service users are directed to root. You'll want messages addressed to root to be sent to your administrative e-mail address. It has the same use as with sendmail. If you choose to use `/etc/aliases`, adjust the `alias_maps` and `alias_database` variables in `main.cf` accordingly.

access Lists allowable networks in this file, including localhost and your LAN; can be in CIDR notation or an appropriate domain name, such as `.example.com` (pay attention to the leading dot, which points to all computers on that domain).

canonical Allows you to substitute one e-mail address for another. See virtual if you've changed domain names of your organization. If you want to use this file, you'll need to enable it by adding the following command to `main.cf`:

```
canonical_maps = hash:/etc/postfix/canonical
```

main.cf Primary Postfix configuration file; generally the only one you'll need to change.

master.cf Lists daemons run by Postfix, as well as conditions such as privileges.

pcre_table If you're familiar with the Perl programming language, you can add Perl expressions for address rewriting or mail routing to this file.

postfix-files Includes a current list of Postfix files and permissions. Don't change the contents of this file.

postfix-script Allows you to administer Postfix with options such as `start`, `stop`, and `reload`.

post-install Not required if you've installed Postfix from a RPM.

regexp_table Lists tables where Postfix looks for accounts; check the current list with the `postconf -m` command.

relocated If users have moved to new locations, enter old and new e-mail addresses on each line.

transport You can specify a certain mail transport protocol such as SMTP or UUCP for the domain of your choice.

virtual You can redirect mail to a specific address or domain to another address in this file.

Example Configuration

To set up Postfix in a basic configuration, we'll modify just the `main.cf` configuration file. To activate or deactivate a command, add or delete the hash mark (#) in front. Open it in the text editor of your choice, and modify the following variables:

1. Activate and change the following variable to point to the name of the local computer (such as `mail.example.com`), which you're configuring as a Postfix server:

   ```
   myhostname = virtual.domain.tld
   ```

2. Specify the network IP address or domain name of the LAN you want to serve. For example, if your domain is example.com and network IP address is 192.168.1.0, you can use *one* of the following commands:

```
mydomain = example.com
mydomain = 192.168.1
```

3. By default, `main.cf` allows Postfix to listen only to the local computer. Activate and deactivate the following commands:

```
#inet_interfaces = all
inet_interfaces = localhost
```

NOTE *If you have a proxy server for your network, you'll have to specify its IP address in the* `proxy_interfaces` *variable.*

4. Finally, use the `mynetworks` variable to specify the LAN and localhost IP addresses in CIDR notation. Here's an example from my LAN:

```
mynetworks = 192.168.1.0/24, 127.0.0.0/8
```

Processing and Activating Postfix

Now you can use the `redhat-switch-mail` tool to make sure Postfix is activated and sendmail is deactivated. Alternatively, you can use the appropriate `service` and `chkconfig` commands to activate and deactivate each service at appropriate runlevels.

If you've configured a firewall, make sure you're allowing SMTP data as a trusted service. Now you can point your e-mail client to the Postfix server.

Using Incoming E-mail Servers

Two basic incoming e-mail servers are in common use today. These servers correspond to the two major incoming e-mail protocols: POP3 and IMAP4. In Red Hat Enterprise Linux, both servers are available as part of the `imap-*` RPM package and are installed as an `xinetd` service (see Chapter 18).

You don't need to create your own e-mail server. You can set up yourself or your clients to use an e-mail server from a provider such as `mail.com` or `yahoo.com`. If you want to create your own e-mail server, install the `imap-*` RPM. Remember to activate its `xinetd` configuration with the `service` ***`servername`*** `on` command and then run `service xinetd reload` to make sure `xinetd` rereads the appropriate configuration file.

If you have a DNS server on your LAN, you can also configure it with an MX entry in the appropriate `/var/named` database file. For more information on DNS, see Chapter 19.

The POP3 protocol is still more popular on the Internet. When you connect from an e-mail client, a POP3 server automatically downloads your e-mail. With most clients, you can choose to keep an original copy of the e-mail on the server.

In contrast, the IMAP4 protocol is more flexible. If you're using an IMAP4 server, you can organize your e-mail on folders on the server. You can search through different messages for keywords, and you can download the messages you want. This is useful for users with multiple computers who need a central database for their e-mail.

NOTE *Red Hat Enterprise Linux also includes support for SquirrelMail, which is a web-based e-mail client package. Naturally, it's installed by default as part of the Mail Server package group.*

The POP3 E-mail Server

Once you've activated a POP3 server, you'll need to create accounts. Anyone who wants to use your POP3 server will require an account on your system. However, those users do not need a home directory.

As you may recall from Chapter 9, the `useradd` *`username`* command automatically creates a home directory for a new user. However, if you add a new user by directly editing `/etc/passwd`, you don't have to add a home directory. Then the `passwd` *`username`* command allows you to assign a new password.

Once you've created a user account, you'll need to tell your user to add the username and the FQDN of the computer that you've configured as the e-mail server to his or her e-mail client. The latter part of this chapter includes details on how to do so with various e-mail clients.

The IMAP4 E-mail Server

After you've activated an IMAP4 server, you'll need to create accounts (the same as you would with a POP3 server). If somebody wants to use your IMAP4 server, that person will need an account on your system. Unlike for a POP3 server, users on an IMAP4 server do need home directories on your system.

The `useradd` *`username`* command automatically creates a home directory for a new user, as detailed in Chapter 9. Then the `passwd` *`username`* command allows you to assign a new password.

When you've created the account, as you would with the POP3 server, you'll need to tell your user to add the username and the FQDN of the computer that you've configured as the e-mail server to his or her e-mail client.

Configuring Mail Clients

Most people use graphical e-mail clients, such as Evolution and Netscape. However, text-based e-mail clients are still popular in the worlds of Linux and Unix. Just as experienced Linux administrators prefer to work from the command-line interface, they often prefer to work with e-mail clients in text mode. While graphical e-mail can be pretty, a graphical e-mail to a large group of users can easily consume the capacity of many e-mail servers.

You may also need to help users configure their own e-mail clients; this should be an easier process.

Text-Based Clients

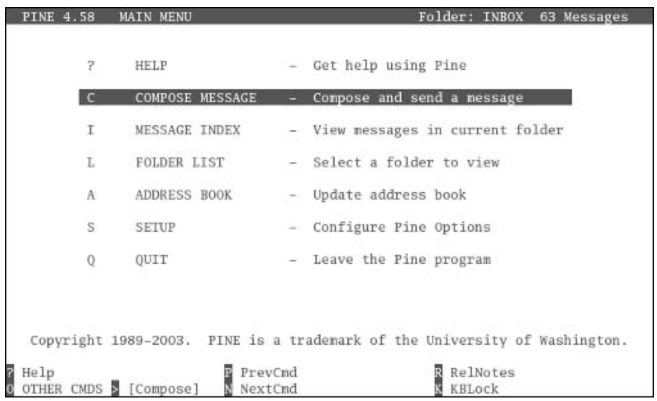
By default, you can use the `mail` program to send and receive e-mail. People who are newer to computers tend to use graphical e-mail clients. However, many users, especially in university and scientific settings, still use text-based e-mail clients. Perhaps the two most common text-based clients are `pine` and `elm`. However, Red Hat Enterprise Linux does not include either of these clients; it does include

the `mutt` text-based e-mail program. While the owner of `pine`, the University of Washington, states that it is released as open source, its license is not completely consistent with the GPL.

Unfortunately, configuring `mutt` for e-mail requires a different paradigm. You can configure a `.muttrc` configuration file in your home directory or modify the generic `/etc/Muttrc` configuration file. The generic file includes several thousand lines and is something we do not cover in this book. Some `mutt` users configure their e-mail accounts through Fetchmail and Procmail configuration files.

As `pine` is still a popular option and is fairly easy to configure, we cover it in this chapter. You can download and compile from the University of Washington at `www.washington.edu/pine`. You can also download and install the `pine` RPM from a third-party source such as `www.rpmfind.net` or `dag.wieers.com/packages/pine`. The first time you run `pine`, you'll see an introduction followed by the main menu displayed in Figure 21.2. Examine the commands in the main part of the screen (?, C, I, L, A, S, and Q) as well as the command options at the bottom of the screen (?, P, R, O, >, N, and K).

FIGURE 21.2
The *pine* main menu



NOTE *The* `pine` *command menu is one place in Linux where case does not matter; for example,* **`P`** *works as well as* **`p`**.

As you can see, you can type **`C`** to start an e-mail message, **`I`** to view current messages, **`L`** to list current folders, and more. But the first step is to set up `pine` to read your e-mail. Type the **`S`** command. You'll see the Setup screen. Next, type **`C`** to begin basic configuration. You'll see a screen similar to Figure 21.3.

While the `pine` setup section includes a large number of configuration options, you'll need to set three things to start receiving your e-mail. These tips assume that your e-mail address is `abcd@example.com` and the incoming e-mail server is `mail.example.com`. Substitute according to your needs.

- Set the `personal-name` to what you want your e-mail recipients to see.
- Add the domain name associated with your e-mail address to `user-domain`. For example, if your e-mail address is `abcd@example.com`, add **`example.com`** to the `user-domain` field.
- Set the `inbox-path`. Based on this example, if `mail.example.com` is a POP3 server, enter **`{mail.example.com/pop3/user=abcd}INBOX`**. If it's an IMAP4 server, enter **`{mail.example.com/user=abcd}`**.

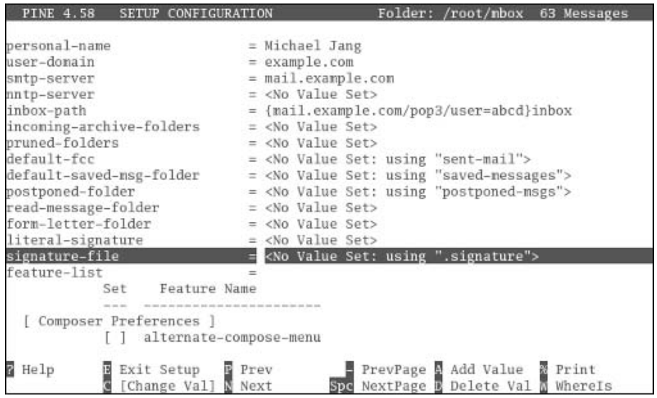


FIGURE 21.3 Basic *pine* configuration

TIP *Some e-mail servers have special requirements. For example, some domains require the full e-mail address, such as* `abcd@example.com`*, as the username in the* `inbox-path`*. They may also require a different domain name for the incoming mail server, the* `user-domain`*. If in doubt, consult your e-mail provider for details.*

If you use an external SMTP server, such as the one associated with your ISP, you can also enter it here. Press the Page Down key a few times. Take a look at the rich variety of options available for `pine`.

NOTE *Before version 4.x,* `pine` *could not handle POP3 e-mail.*

When you're through making changes, type **E** to exit the Setup Configuration screen. Assuming you're satisfied with the changes, type **Y** to confirm when prompted; you'll be taken back to the main menu shown in Figure 21.2. In the main menu, type **L**, and select INBOX. The first time you do this, you should be prompted for your e-mail password.

Next, `pine` will go to your e-mail server, get your latest messages, and show them to you in a message index similar to Figure 21.4 (I've masked the identity of my e-mails). From here, everything is fairly intuitive, and you can use the command options shown at the bottom of the screen. To read a message, highlight it and press Enter.

Creating a new message is easy. Return to the main menu as shown in Figure 21.2; then type **C** to start composing a new message. If you've ever used e-mail before, the format shown in Figure 21.5 should be quite familiar. The commands at the bottom are Control characters; for example, when you're done with a message, the Ctrl+X command sends your message (after you type **Y** to confirm). If you've configured an SMTP server or your sendmail service is working, `pine` should send your message automatically.

Graphical Clients

Three basic graphical e-mail clients are available in Linux: Evolution, Mozilla Mail, and KMail. While we describe these clients briefly in Chapter 30, we'll look at the configuration windows for the Evolution client in this section.

FIGURE 21.4
The pine message index

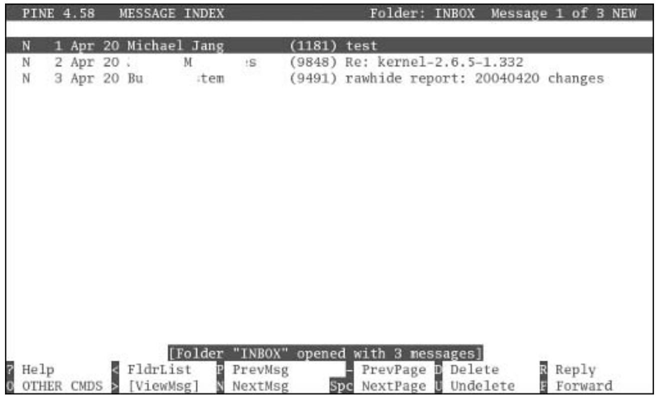
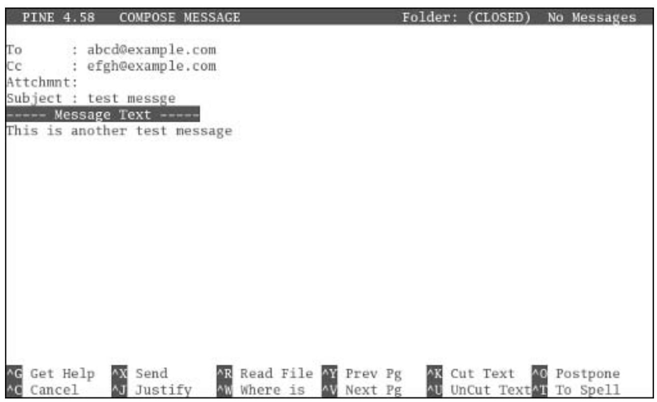


FIGURE 21.5
Creating *pine* e-mail



For any e-mail client, the configuration requirements are the same. As you may have done with the `pine` text e-mail client, you'll need at least the information described in Table 21.2.

The configuration process for all three clients is elementary for the target audience for this book. We've illustrated Evolution just for your reference.

TABLE 21.2: DATA NEEDED FOR E-MAIL CLIENTS

DATA	DESCRIPTION
Name	The name you want other users to see
Domain Name	The information after the @ in your e-mail address
Inbox Server	The FQDN of your incoming e-mail server; sometimes listed as "Host" or "Server Name"
Username	Your username or login on the e-mail server; normally just the part before the @ in your e-mail address

Configuring Evolution

If you're using Evolution, open it in your favorite GUI. From the menu bar, select Tools ➢ Settings. This opens an Evolution Settings window, as shown in Figure 21.6.

In the left pane, scroll up until you can click the Mail Accounts icon; then click Add to access the Evolution account wizard, or highlight an existing account and click Edit. This should bring you to the Evolution Account Editor. Enter your basic account information on the Identity tab, and then click the Receiving Mail tab shown in Figure 21.7. Select a Server Type if required, enter the FQDN for your inbound e-mail server in the Host text box, and enter your username on that server in the Username text box.

Figure 21.6
Evolution Settings window

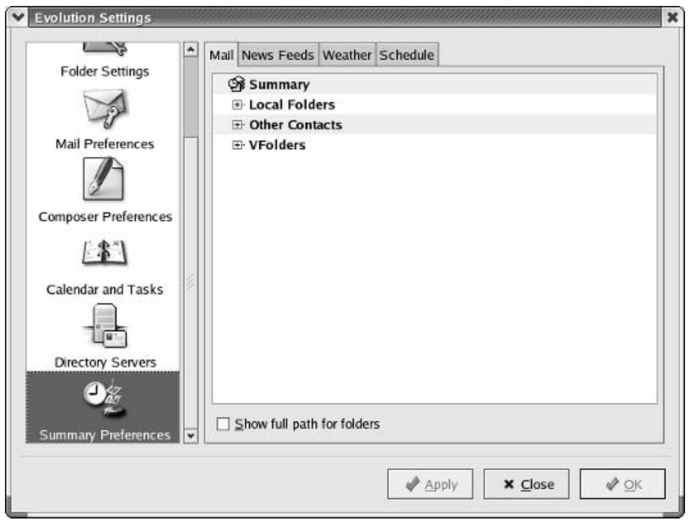


Figure 21.7
Configuring an Evolution e-mail account

Summary

There are servers that send e-mail, and servers that receive e-mail. Modern versions use some basic TCP/IP protocols: SMTP, POP3, and IMAP4. To send and receive e-mail through these protocols, you can choose among three types of mail services: MTA, MDA, and MUA. An MTA such as sendmail sends e-mail through a network. An MDA such as procmail takes messages from the Internet and stores them in spools, sometimes on incoming e-mail servers. An MUA is an e-mail client such as `pine`, `mutt`, or Evolution.

sendmail is currently the most popular outgoing e-mail server on the Internet. Since editing the `sendmail.cf` configuration file is difficult, Red Hat provides a macro file, `sendmail.mc`, which can be more easily understood and edited. It is easy to convert into `sendmail.cf` with the `m4` macro processor. There are other important sendmail configuration files, including `/etc/sysconfig/sendmail` and `/etc/aliases`, as well as other files in the `/etc/mail` directory. Once you have your new `sendmail.cf` file, you can make the sendmail daemon reread it with the `service sendmail restart` command.

Postfix is an alternative e-mail server that may be easier to configure. There are a substantial number of configuration files in the `/etc/postfix` directory. However, all you need to do to configure Postfix for your system is modify four variables in `/etc/postfix/main.cf` and make sure that Postfix is enabled and that sendmail disabled at appropriate runlevels. This is easy to do with the `chkconfig` and `service` commands or the `redhat-switch-mail` tool.

There are two basic options for e-mail servers that conform to the POP3 and IMAP4 protocols. Secure versions of each server are available. All are `xinetd` services that can be installed from the `imap-*` RPM package. Once these services are installed and activated, your users will need a username and the FQDN of the e-mail server. If it's an IMAP4 server, they'll also need a home directory for their e-mail files.

Both text and graphical e-mail clients are available. One useful highly configurable text-based client is `pine`. Graphical e-mail clients are available in a number of forms, including Evolution, Mozilla Mail, and KMail.

In the next chapter, we'll take a look at various FTP clients and servers. The FTP client is flexible; you can even use FTP commands to connect and upgrade your RPMs. You can install anonymous, standard, and even secure FTP servers on your Red Hat Enterprise Linux computer.

Part 6

Linux File Sharing Services

In this Part, you will learn:

- Chapter 22: Linux Sharing Services: FTP and NFS
- Chapter 23: Linux Authentication Services: NIS and LDAP
- Chapter 24: Making Samba Work for You
- Chapter 25: Web Services
- Chapter 26: Setting Up MySQL for Databases

Chapter 22

Linux Sharing Services: FTP and NFS

ON A NETWORK WITH Linux and Unix computers, two common sharing services are the File Transfer Protocol (FTP) and Network File System (NFS). FTP is one of the oldest members of the TCP/IP protocol stack, yet it is still in common use today. As the name suggests, it is optimized for transferring files. NFS lets you mount remote directories seamlessly on your Linux computer. NIS allows you to keep a common database of key configuration files on your network.

There are FTP clients and FTP servers. A rich variety of commands are associated with FTP clients; you can even upgrade RPMs directly with the right `ftp` command. And of course, GUI FTP clients exist that work just as well.

Many FTP servers are available for Linux, and in this chapter we cover two of them: Very Secure FTP (vsFTP) and Washington University's (St. Louis) WU-FTP. Both servers can be configured to allow anonymous users. vsFTP is now the default; while now unsupported by Red Hat, WU-FTP is still a popular alternative.

When you mount an NFS directory, you may not be able to tell the difference from a directory on your own computer. For example, you could configure home directories for all your users on a server and share them through NFS. Then you could configure client computers on your LAN to mount `/home` during the boot process. NFS may look a bit complex because it uses up to six daemons, but the basic configuration files and commands are easy. If you're less familiar with NFS, the graphical `redhat-config-nfs` tool can help. And in this chapter, you'll learn to understand and manage the risks commonly associated with NFS. This chapter covers the following topics:

- Using FTP as a client
- Configuring the Very Secure FTP server
- Configuring WU-FTP with real users
- Creating an anonymous FTP server
- Configuring Network File System servers
- Configuring with *redhat-config-nfs*
- Working with NFS clients

Using FTP as a Client

The FTP service has a long history, with commands that predate shells such as bash. You should learn how to use FTP as a client, at least because key Red Hat RPMs are updated on FTP servers. As with other Linux clients, GUI FTP clients such as gFTP (GNOME FTP) are simply "front ends" for the commands that you can run at the text console.

The following sections describe a connection from an FTP client to Red Hat's main FTP site, `ftp.redhat.com`. This site is often quite busy, especially during working hours in the United States. Red Hat has a list of a large number of FTP mirror sites (`www.redhat.com/download/mirror.html`) that should include files that are nearly as up-to-date as those you'll find at `ftp.redhat.com`. If you have problems accessing `ftp.redhat.com`, try one of the mirror sites.

NOTE *You can only get Red Hat Enterprise Linux source RPMs from the Red Hat FTP site. If you want the official binary RPMs, you'll need an account that comes with an official subscription to this operating system. Alternatively, you could use one of the FTP sites that carry the third-party rebuilds of Red Hat Enterprise Linux as described in Chapter 1.*

Basic Commands

As you can see in Figure 22.1, a substantial number of commands are associated with the FTP client. This section covers only an essential few FTP commands; data on even rarely used FTP commands is available through the FTP manual you can access with the `man ftp` command. You can view a simple description of a command from the `ftp>` prompt by entering **`help`** ***`command`***.

FIGURE 22.1
FTP client commands

```
230-welcome to Mike's FTP server
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated.  Commands are:

!               cr              mdir            proxy           send
$               delete          mget            sendport        site
account         debug           mkdir           put             size
append          dir             mls             pwd             status
ascii           disconnect      mode            quit            struct
bell            form            modtime         quote           system
binary          get             mput            recv            sunique
bye             glob            newer           reget           tenex
case            hash            nmap            rstatus         trace
ccc             help            nlist           rhelp           type
cd              idle            ntrans          rename          user
cdup            image           open            reset           umask
chmod           lcd             passive         restart         verbose
clear           ls              private         rmdir           ?
close           macdef          prompt          runique
cprotect        mdelete         protect         safe
ftp> help rmdir
rmdir           remove directory on the remote machine
ftp> help open
open            connect to remote ftp
ftp>
```

Table 22.1 describes some important FTP commands. You may note similarities between a number of these commands and those you know in the bash shell.

TABLE 22.1: BASIC FTP CLIENT COMMANDS

COMMAND	**DESCRIPTION**
`!`*`command`*	Runs a shell command on the local computer, in the local directory.
`ascii`	Sets file transfer to ASCII mode; best for text files.
`binary`	Sets file transfer to Binary mode; best for executables and compressed files.
`bye`	Exits from the current FTP session; synonym for `exit`.
`cd`	Changes the directory; similar to the Linux version of this command.
`dir`	Equivalent to the `ls -l` shell command.
`get` *`ftpfile localfile`*	Copies the *`ftpfile`* file from the FTP server to *`localfile`* on the local computer; `mget` allows you to use wildcards, which is also known as *globbing*.
`ls`	See `dir`.
`put` *`localfile ftpfile`*	Copies the *`localfile`* file from the local computer to *`ftpfile`* on the FTP server; `mput` allows you to use wildcards/globbing.
`pwd`	Lists the current working directory on the FTP server; if you've configured FTP securely, the root directory that you see on the FTP server will be the main directory for FTP files, usually `/var/ftp`.
`user`	Allows you to enter a username; prompts for a password.

Connecting to *ftp.redhat.com*

Now let's get some practice using the command-line FTP client. Assuming your Linux computer is connected to the Internet, run the `ftp ftp.redhat.com` command. The Red Hat FTP site allows only anonymous connections. While the commands shown in Figure 22.2 seem to require a password, no special password is needed. By custom, when you connect to an FTP server anonymously, you're supposed to enter your e-mail address when prompted for a password.

NOTE *You can set up an FTP connection on your own network. We describe two different FTP servers in this chapter. Once the server is active, you can connect to it from the local computer with the* `ftp localhost` *command.*

FIGURE 22.2
Connecting to an FTP server

```
[root@Enterprise3 root]# ftp ftp.redhat.com
Connected to ftp.redhat.com.
220 Red Hat FTP server ready. All transfers are logged. (FTP)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (ftp.redhat.com:root): anonymous
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

At the `ftp>` prompt, enter the commands you need. Try out some of the commands shown in the previous section. You may note that commands such as `put` do not work; anonymous users aren't allowed to write to standard Red Hat FTP servers.

NOTE *By default, the root user is not allowed to access any FTP server. If you try to log in through FTP as root, even a correct password will be rejected.*

As an example, navigate to the directory with i386 Red Hat Enterprise source RPMs. As of this writing, they are located in the `/pub/redhat/linux/enterprise/3/en/os/i386/SRPMS` directory. You should find a long list of source RPMs here.

If you don't have a subscription to Red Hat Enterprise Linux, you can still download, compile, and install source RPMs. The commands and directories you need are described in Chapter 10. Download the packages from the current Enterprise source package database, using commands similar to those shown in Figure 22.3. You can then compile and install or upgrade these packages at your leisure.

FIGURE 22.3
Downloading a source RPM

```
227 Entering Passive Mode (66,187,224,51,233,118)
150 Here comes the directory listing.
226 Directory send OK.
ftp> cd SRPMS
250 Directory successfully changed.
ftp> ls y*
227 Entering Passive Mode (66,187,224,51,233,120)
150 Here comes the directory listing.
-rw-r--r--    6 ftp      ftp        528485 Oct 21  2003 yelp-2.2.3-1.E.src.rpm
-rw-r--r--    6 ftp      ftp        169845 Oct 21  2003 yp-tools-2.8-1.src.rpm
-rw-r--r--    6 ftp      ftp        154625 Oct 21  2003 ypbind-1.12-1.src.rpm
-rw-r--r--    6 ftp      ftp        162159 Oct 21  2003 ypserv-2.8-1.src.rpm
226 Directory send OK.
ftp> mget ypbind*
mget ypbind-1.12-1.src.rpm? y
227 Entering Passive Mode (66,187,224,51,233,128)
150 Opening BINARY mode data connection for ypbind-1.12-1.src.rpm (154625 bytes)
.
226 File send OK.
154625 bytes received in 2.7 seconds (56 Kbytes/s)
ftp> bye
221 Goodbye.
[root@Enterprise3 root]#
```

The GUI FTP Client

Of course, there are graphical versions of the FTP client. One common graphical FTP client is gFTP, which you can start by entering `gftp` in a command-line interface in your favorite GUI. This opens the gFTP client, shown in Figure 22.4.

NOTE *You can run* `gftp` *from a regular virtual console; it's part of the gFTP package and opens a text-mode FTP client similar to* `ftp`.

The gFTP client is convenient; it has several common sites preconfigured in the Bookmarks menu. However, the sites, such as what you may open when you select Bookmarks ➤ RedHat Sites ➤ Freshmeat RPMs, aren't always kept up-to-date. Nevertheless, it is a convenient way to make an FTP connection. For example, try Bookmarks ➤ RedHat Sites ➤ RH Main. If the Red Hat FTP server is not overloaded, it should bring you to the base Red Hat FTP directory shown in Figure 22.5.

FIGURE 22.4
The gFTP client

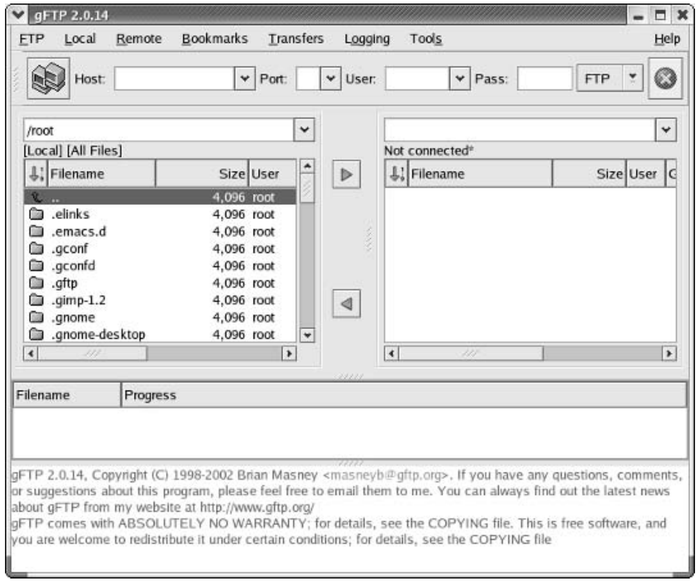
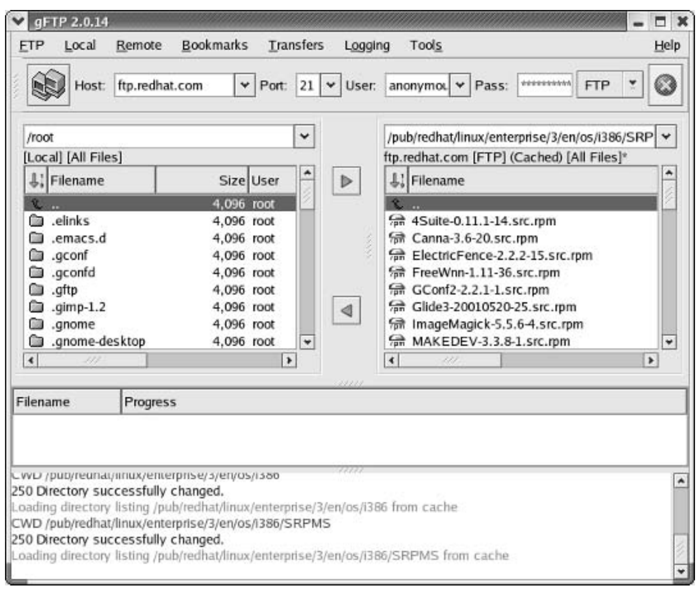


FIGURE 22.5
gFTP in action



Compare the differences between Figure 22.4 and Figure 22.5. When you select a gFTP bookmark, it fills in a number of entries (described in Table 22.2) in various text boxes.

TABLE 22.2: ENTRIES FOR CONNECTING A GFTP CLIENT

ENTRY	FUNCTION
Host	The FQDN of the FTP server.
Port	The TCP/IP port for the connection; by default, it's 21.
User	The username for the connection; anonymous is common for an anonymous FTP server.
Pass	The password associated with the username. By convention, it's supposed to be your e-mail address.

You may want to go to a subdirectory on an FTP server. To navigate to the desired directory, double-click it. Remember, the double-dot (`..`) is associated with the next higher-level directory. Once you've found the folder or file you want to copy, highlight it and click the arrow button.

You can observe the commands and messages that are transmitted in the bottom part of the gFTP window.

Configuring the Very Secure FTP Server

One of the big drawbacks of a regular FTP server is security. One compromise is the Very Secure FTP server (vsFTP). Red Hat has made it the default and only FTP server for Red Hat Enterprise Linux 3.

While it does not encrypt communications, vsFTP does avoid some of the security problems commonly associated with WU-FTP. It's used as a standard FTP server for a number of sites, including `ftp.redhat.com`. It can be configured for anonymous or real users. In fact, the home page for this server (`http://vsftpd.beasts.org`) suggests it is faster than WU-FTP.

vsFTP shares a number of characteristics with WU-FTP. Where possible, we'll describe these characteristics in the sections that follow.

Basic Security Features

The commands associated with vsFTP are normally configured with minimal privileges; this reduces the risk of a cracker using one of these commands to gain root access to your system.

Configuration Files

The vsFTP package includes configuration files in the `/etc` directory. Two of these files, `vsftpd.ftpusers` and `vsftpd.user_list`, essentially disallow access from privileged users. This list is simple; it includes a list of users, such as root, bin, and adm. The main configuration file is `/etc/vsftpd/vsftpd.conf`. The following is a line-by-line analysis of the default configuration file, which includes several options. More details are available via the `man vsftpd.conf` command.

We've included the entire file with our own comments to help you understand each command.

```
# Example config file /etc/vsftpd.conf#
# The default compiled in settings are very paranoid. This sample file
```

```
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=YES
```

The first notes, starting with the #, are comments. You may have noticed that the first comment line is wrong; it reflects the old location of the `vsftpd.conf` file. By default, vsFTP allows anonymous access with the previous command. Users can log anonymously as user anonymous or ftp.

```
# Uncomment this to allow local users to log in.
local_enable=YES
```

The default Red Hat configuration `local-enable` variable allows users with a regular account on the FTP server to log in as real users.

```
# Uncomment this to enable any form of FTP write command.
write_enable=YES
```

These users have access to all directories on the FTP server, including the root (`/`) directory. You may want to comment out the `write_enable` command; otherwise, logged-in users have a dangerous level of access to your system. You can also configure all access to an unprivileged user, as described later with the `nopriv_user` variable.

To minimize the problem, you could add the `chroot_local_user=YES` command, which prevents users from accessing the root (`/`) directory on the FTP server. However, users who are allowed to upload to their home directories could then upload executable files that may compromise the security of the server.

The default Red Hat configuration allows real users to delete files in their home directories. It does not allow anonymous users to delete files.

```
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
```

Without this `local_umask` command (see `umask` in Chapter 6), uploaded files have read and write permissions, limited to the owner of the file. With this command, all users have at least read permissions to uploaded files.

```
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
```

Sometimes, you want to allow anonymous users to upload to your FTP server. While you risk having users overload the partition with the `/var` filesystem, you can limit this risk by mounting `/var` on a separate partition, as discussed in Chapter 7. As we describe later in our discussion of anonymous servers, you'll need to set appropriate permissions for the directory where you accept uploads, such as `/var/ftp/pub`. Note that this setting is disabled in the default `vsftpd.conf` configuration file.

```
# Uncomment this if you want the anonymous FTP user to be able to create
```

```
# new directories.
#anon_mkdir_write_enable=YES
```

You can also let anonymous users create new directories wherever they have write permissions. Note the comment mark (#) in front of the command, which disables the setting; you'd also need to add an `anon_other_write_enable=YES` line to let users actually write files to the new directories.

```
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
```

By default, users are allowed to see messages in a `.message` file in different directories. When the user changes to that directory, the contents of the relevant `.message` file (or a filename specified by the `message_file=`*`filename`* command) are shown.

```
# Activate logging of uploads/downloads.
xferlog_enable=YES
```

Normally, a record of uploads and downloads are stored in `/var/log/vsftpd.log`. You can specify a different file with the `xferlog_file=`*`filename`* command.

```
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
```

Some FTP clients may require the previous command. Port 20 is one of the TCP/IP ports shown in `/etc/services`.

```
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
```

The user who uploads a file does not have to own that file. For example, the following commands, slightly different from what you see in the `vsftpd.conf` file, would change ownership of any uploaded files to user mj:

```
chown_uploads=YES
chown_username=mj
```

Next, we look at the standard log file location:

```
# You may override where the log file goes if you like. The default is shown
# below.
#xferlog_file=/var/log/vsftpd.log
```

Normally, vsFTP log files are stored in `/var/log/vsftpd.log`. You can change this to the location of your choice.

```
# If you want, you can have your log file in standard ftpd xferlog format
xferlog_std_format=YES
```

This command enables the standard format for logging uploads and downloads to the FTP server, as used for WU-FTP. Try disabling this command by adding # in front. Write this file, and then set up a transfer from an FTP connection. Read the results in the /var/log/vsftpd.log file. The non-standard vsFTP log format is more descriptive.

```
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
```

The vsFTP server regulates how long a user can sit idle while logged on. By default, it's 300 seconds. The previous command, if active, changes this period to 10 minutes.

```
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
```

Sometimes there are errors during a file transfer. If there is an error, the FTP client will try to reconnect. But there comes a point where it is better to restart the connection. The default period is 300 seconds; the previous command, if active, changes that to two minutes.

```
t# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
```

You can set up a special unprivileged user, ftpsecure, by enabling the previous command. If you do, make sure the user exists in /etc/passwd.

Setting Up Special FTP Users

You should set up ftpsecure almost as a "guest" user. Once configured, all users who connect to your FTP server get the ftpsecure username. If you don't want users to log in directly to your computer, you can change the associated entry in /etc/password to set up /sbin/nologin as the default shell:

```
ftpsecure:x:601:601::/home/ftpsecure:/sbin/nologin
```

The following command allows less-capable FTP clients to cancel a download without hanging:

```
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
```

However, this setting is not needed for the regular command-line FTP client described earlier in this chapter.

```
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that turning on ascii_download_enable enables malicious remote parties
# to consume your I/O resources, by issuing the command "SIZE /big/file" in
# ASCII mode.
```

```
# These ASCII options are split into upload and download because you may wish
# to enable ASCII uploads (to prevent uploaded scripts etc. from breaking),
# without the DoS risk of SIZE and ASCII downloads. ASCII mangling should be
# on the client anyway..
#ascii_upload_enable=YES
#ascii_download_enable=YES
```

If you need to transfer files in ASCII mode, enable one or both of the previous "`ascii`" commands. It should rarely be necessary, even for text files, unless you need to preserve certain types of formatting.

```
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
```

You can configure the previous `ftpd_banner` message for users before they log in. For example, you could change the message as follows to encourage anonymous logins:

```
# ftp_banner=Welcome. Type ftp at the prompt for an anonymous login.
```

Sometimes crackers will attempt something similar to the "ping of death" described in Chapter 17 on your FTP server.

```
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
```

If you enable both of the previous commands, you can create a list of anonymous passwords in `/etc/vsftpd.banned_emails` that aren't allowed access. This can deny access to crackers who use automated tools to try to bring down your FTP server.

```
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
```

If you activated `chroot_list_enable=YES`, you can configure a group of users who see their home directory as the root (`/`) directory in `/etc/vsftpd.chroot_list`. If you also configure `chroot_local_user=YES`, the effect of the list in `/etc/vsftpd.chroot_list` is reversed.

```
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
```

If you activate the previous command, FTP clients can run the `ls -R` command on any available directory, which allows users to see the contents of subdirectories. However, this is disabled by default; a user who is logged into an FTP server multiple times could create a large load by running `ls -R` on all sessions.

```
pam_service_name=vsftpd
```

The `pam_service_name` lists the Pluggable Authentication Module (PAM) file associated with vsFTP. For more information on PAM, see Chapter 17.

```
userlist_enable=YES
```

This command makes vsFTP check for prohibited usernames in the `/etc/vsftpd.user_list` file.

```
#enable for standalone mode
listen=YES
```

This allows vsFTP to be run as its own daemon, supported by the `vsftpd` script in the `/etc/rc.d/init.d` directory. Otherwise, you could run vsFTP as an `xinetd` script described in Chapter 18.

```
tcp_wrappers=YES
```

Finally, the `tcp_wrappers` command allows you to regulate access through the TCP Wrappers firewall through the `/etc/hosts.allow` and `/etc/hosts.deny` files. We described these files in more detail in Chapter 18.

Configuring WU-FTP with Real Users

The information in the following and the previous sections is based on the WU-FTP server package, which must now be loaded from a third-party site. We've already described how to enable anonymous user access. In this section, you'll learn about the configuration files associated with WU-FTP and how to apply them to regular users on your system.

NOTE *The latest available RPM package for WU-FTP is based on Red Hat Linux 8.0. I've downloaded and installed it on my Red Hat Enterprise Linux 3 system. Alternatively, you can download, unpack, and install the latest "tarball" from* `ftp.wu-ftpd.org`*, using the techniques described in Chapter 12.*

Configuration Files

Several configuration files are associated with the WU-FTP package, all in the `/etc` directory: `ftpaccess`, `ftpconversions`, `ftpgroups`, `ftphosts`, and `ftpusers`.

Of these files, `ftpusers` is now obsolete and `ftpgroups` is rarely used; the functionality of these files is now part of `ftpaccess`. In this section, we describe the other configuration files in detail.

Alternate examples of each of these configuration files are available in the WU-FTP documentation, in the `/usr/share/doc/wu-ftpd-`*`versionnumber`*`/examples` directory.

We examined a couple of characteristics of the default `/etc/ftpaccess` file earlier in this chapter. Now it is time to examine this file line by line. The first lines take the functionality of `/etc/ftpusers`.

```
deny-uid %-99 %65534-
deny-gid %-99 %65534-
allow-uid ftp
allow-uid ftp
```

These lines deny access to User and Group IDs less than 99 and greater than 65534, except user ftp. If you examine your `/etc/passwd` and `/etc/group` files, you'll see that these ID numbers are associated with administrative accounts. You can limit access to all users except ftp with the following simple change:

```
deny-uid *
deny-gid *
allow-uid ftp
allow-uid ftp
```

The following line sets up the chroot jail. All users are classified as guest users, and they're limited to their home directories. For example, if user mj logs in, he is sent to `/home/mj`, and is unable to access higher-level directories.

```
guestuser *
```

We'll discuss the next line later in this chapter; user mj isn't allowed to navigate to the `/home` or root (`/`) directory unless the following line is activated:

```
# realuser user1,user2
```

Remember, the hash mark (`#`) makes Linux ignore the information that follows; if you remove `#`, *`user1`* and *`user2`* gain full user privileges on that FTP server. The following line can be used to limit the users on the `realuser` list. For example, if the previous line was `realuser *`, you can add the `ftpchroot` group to `/etc/group`. Members of the `ftpchroot` group would not be allowed to navigate above their respective home directories:

```
# guestgroup ftpchroot
```

NOTE *The management of user and group configuration files such as* `/etc/passwd` *and* `/etc/group` *is discussed in Chapter 9.*

As described earlier, the first line that follows allows access from real, guest, and anonymous users. The next line, if active, limits access to real users who log in from the 192.168.0.0/24 network. Anonymous access is not allowed; users need to enter their passwords. One obvious drawback is that real user passwords are sent over your LAN in clear text.

```
class    all    real,guest,anonymous    *
# class    all    real    192.168.0.0/24
```

If you comment out the previous `guestuser *` line, you can substitute *real* for *guest*.

```
class    all    guest   192.168.0.0/24
```

If you're the administrator for your server, you'll want to substitute your e-mail address here.

```
email root@localhost
```

The following command limits the number of attempted logins. In this case, after five login attempts this FTP server closes the connection.

```
loginfails 5
```

In the Linux and Unix worlds, `README*` files are commonly used for instructions or to supply more information about the packages contained in a specific directory. The following lines return a `Please read the file README` message whenever a user logs into and changes to a directory with a README file:

```
readme    README*    login
readme    README*    cwd=*
```

As the administrator of the FTP server, you may want to send other messages to your users. The following lines allow you to add a welcome message to the `welcome.msg` file in the opening directory. You can also add `.message` files to send additional messages to users who use the `cd` command to navigate to those directories.

```
message    /welcome.msg    login
message    .message        cwd=*
```

You can see what happens when I added a README file to the `/var/ftp` directory as well as information to various message files in Figure 22.6.

FIGURE 22.6
FTP login messages

```
[root@Enterprise3d root]# ftp Enterprise3
Connected to Enterprise3.
220 Enterprise3 FTP server (Version wu-2.6.2-8) ready.
504 AUTH GSSAPI not supported.
504 AUTH KERBEROS_V4 not supported.
KERBEROS_V4 rejected as an authentication type
Name (Enterprise3:root): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-The response 'fsalj' is not valid
230-Next time please use your e-mail address as your password
230-    for example: joe@192.168.1.4
230-welcome to Mike's FTP server
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd pub
250-welcome to Mike's FTP sub-server
250-
250 CWD command successful.
ftp>
```

It's useful to store packages in compressed format on an FTP server. The following commands allow users who access such packages to have them uncompressed or unpackaged automatically, per the commands in `/etc/ftpconversions`, which is described in a later section, "*/etc/ftpconversions*":

```
compress    yes    all
tar         yes    all
```

Access to dangerous commands can also be limited. By default, `/etc/ftpaccess` limits access to four commands, as shown. If you keep the `guestuser *` line and modify `/etc/ftpaccess` slightly (as shown in bold), all users who log into your FTP server are either guest or anonymous users. You can then prevent all users from using these commands.

```
chmod     no    guest,anonymous
delete    no    guest,anonymous
```

```
overwrite  no    guest,anonymous
rename     no    guest,anonymous
```

While logins to the FTP server are normally stored in `/var/log/messages`, file transfers to and from the server are logged to `/var/log/xferlog`.

```
log transfers anonymous,guest,real inbound,outbound
```

If you run the `ftpshut` command, it creates a temporary `/etc/shutmsg` file. This command refuses additional logins if a shutdown of the FTP server is imminent:

```
shutdown /etc/shutmsg
```

Anonymous users are supposed to enter their e-mail addresses as the password. If they do, you can see their passwords in `/var/log/messages`. The following command sends a warning to users who connect to the FTP server without entering an e-mail address in proper format. As configured, users are still logged onto the server even with an invalid e-mail address.

```
passwd-check rfc822 warn
```

Limits in */etc/ftpaccess*

If you're running an FTP server on the Internet, you may want to limit the number of simultaneous users connected to your server. This can help ration the speed at which your users can download their files.

One simple way is to add another command to `/etc/ftpaccess` with the `limit` command. For example, the following command prevents more than 20 users from signing onto your FTP server at any one time. The `warning.msg` file is sent to users who try to log in when the limit is reached.

```
limit    all    20    any     warning.msg
```

Perhaps you just want to limit access to users during the day (8 a.m. to 5 p.m.), when your server may be busy with other tasks.

```
limit    all    20    Wk0800-1700    warning.msg
```

The syntax of time in this command is based on the UUCP remote host description file. The easiest way to find this file is by searching for `l.sys` in your favorite search engine.

Tip *I like to search the newsgroups for answers to common Linux problems. Remember, Linux is under constant development by a worldwide community of users and developers; they often discuss their Linux issues through newsgroups and many other forums. It's easy to search through the newsgroups via* `groups.google.com`*.*

You can also limit the amount of data that a user can download from your FTP server. For example, the following command limits the amount of downloadable files to 100MB:

```
byte-limit    out    100000000    all
```

Alternatives to `out` (downloads) are `in` (uploads) and `total` (both directions).

/ETC/FTPCONVERSIONS

The `/etc/ftpconversions` file, shown in Figure 22.7, allows you to run selected commands during the upload or download process. For example, if you have a compressed file of pictures named `pictures.gz` on your FTP server, the third line in `/etc/ftpconversions` lets you download and uncompress the pictures directly with the following command at the `ftp>` prompt:

```
ftp> get pictures
```

Note how the `.gz` is left out of the request. The FTP server automatically refers to `/etc/ftpconversions` for the needed command.

FIGURE 22.7
/etc/ftpconversions

```
:.Z:  :  :/usr/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
:   : :.Z:/usr/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
:.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
:   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
:   : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
:   : :.tar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
:   : :.tar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
~
~
~
~
"/etc/ftpconversions" 7L, 464C                          1,2          All
```

/ETC/FTPHOSTS

The `/etc/ftphosts` file looks conceptually similar to the `/etc/hosts.allow` and `/etc/hosts.deny` files associated with `xinetd` services (see Chapter 18). You can allow and deny access to the FTP server from specific users. However, the functionality isn't quite what you may expect.

For example, the following line allows FTP access only from user bbonds from the computer with the given IP address. No other users and no other computers are allowed access to this FTP server. You can substitute the FQDN for the IP address.

```
allow    bbonds     192.168.0.32
```

Alternatively, the following line denies access to user wmays only from the noted computer:

```
deny    wmays     linux.example.com
```

Commands

FTP server commands let you regulate when FTP servers are active and allow you to view a list of currently connected users. For example, the following command warns users at their next command that the FTP server will shut down in 15 minutes, or at 3:30 p.m.:

```
ftpshut +15 "The FTP Server will close in 15 minutes"
ftpshut 1530 "The FTP server will stop at 3:30 PM"
```

You can set this up as a `cron` script, as discussed in Chapter 13. This allows you to shut down the FTP server on a regular basis. Other FTP server–related commands are listed in Table 22.3.

TABLE 22.3: FTP SERVER COMMANDS

COMMAND	DESCRIPTION
`ftpwho`	Lists connected users and origin IP addresses
`ftpcount`	Lists number of connections
`ftpshut`	Allows you to shut down an FTP server now or at a specified time
`ftprestart`	Stops and restarts an FTP server

Anonymous Uploads

By default, anonymous users aren't allowed to write to any of the `/var/ftp` directories. In some cases, you may want to allow users to supply their files in a directory such as `/var/ftp/pub`.

To allow uploads, you'll need to modify the `/etc/ftpaccess` file and the permissions on the appropriate directory. For example, the following line allows uploads to the `/var/ftp/letter` directory:

```
upload    /var/ftp    /letter    yes    cindy    ywow    0660
```

On the FTP server, these files are owned by user cindy, group ywow, with 660 permissions that allow the user cindy and members of the ywow group to read and write to uploaded files.

You'll also need proper permissions on the upload directory. To write a file to a directory, you need at least write and execute permissions. In this case, the `chmod 733 /var/ftp/letter` command would meet these minimum requirements. Of course, if you want regular users on the server to read the files in that directory, you can provide less restrictive permissions with a command such as `chmod 733 /var/ftp/letter`. For more information on permissions, see Chapter 6.

Creating an Anonymous FTP Server

It is not difficult to create an anonymous FTP server. However, there are details involved in securing that server. When the server is properly configured, users won't be able to get above the base FTP directory, `/var/ftp`, and certainly not to the root (`/`) directory. The default Red Hat FTP configuration is based on the vsFTP server.

The following sections show you how to create a basic anonymous-only FTP server. It can work with vsFTP or WU-FTP servers. You can customize the configuration further using many of the settings described earlier in this chapter.

Configuring vsFTP

Once the appropriate packages are installed, you'll need to activate the service. Assuming you're using vsFTP, you'd run the `service vsftpd start` command. Remember to use the appropriate `chkconfig` command (see Chapter 13) to make sure vsFTP is active the next time you start Linux.

As discussed earlier, the vsFTP configuration file, `vsftpd.conf`, allows anonymous access by default. To keep it that way, you need to watch the following two commands in that file:

```
anonymous_enable=yes
#local_enable=yes
```

Naturally, if you want to limit access to anonymous users, you'll want to enable the `anonymous_enable` command and disable the `local_enable` command.

Configuring WU-FTP

If you've installed the WU-FTP server, you'll need to work with several `/etc/ftp*` configuration files, as described in the following sections. The next major section, "Configuring WU-FTP with Real Users," describes each configuration file in more detail.

NOTE *WU-FTP is no longer included with Red Hat Enterprise Linux, but you can download it from the FTP site at* `ftp.wu-ftpd.org` *or the SpeakEasy RPM library at* `www.rpmfind.net`*. In this chapter, I've downloaded and installed the Red Hat Linux 8 version of WU-FTP, which includes the latest available RPM as of this writing.*

Setting Up Anonymous Directories

You can set up a basic anonymous FTP connection on WU-FTP. You'll need the `anonftp-*` RPM to install several subdirectories in `/var/ftp` for the files and commands that an FTP user needs to navigate in that directory and its subdirectories. These subdirectories are listed in Table 22.4.

TABLE 22.4: ANONYMOUS FTP DIRECTORIES

DIRECTORY	DESCRIPTION
/var/ftp/bin	Executable shell commands; available commands are limited.
/var/ftp/etc	Configuration files; by default includes abbreviated versions of passwd and group.
/var/ftp/lib	Program libraries.
/var/ftp/pub	Files for users; permissions can be configured for uploads.

You need to know that WU-FTP is an `xinetd` service; the techniques described in Chapter 18 apply. Make sure that the service is not disabled in the `/etc/xinetd.d/wu-ftpd` file and that it isn't blocked in `/etc/hosts.deny` (as well as by any `iptables` firewall that may be active).

RESTRICTING ACCESS

It's easy to limit access to an FTP server to anonymous users. First, open the `/etc/ftpaccess` configuration file. By default, it should include the following entry:

```
# User classes . . .
class    all    real,guest,anonymous    *
```

This FTP access `class` allows access to real, guest, and anonymous users from all addresses. Limit access to anonymous users from the 192.168.0.0/24 network by changing this line as follows:

```
class    all    anonymous    192.168.0.0/24
```

Setting Up Anonymous FTP Security

There are several default measures that protect an anonymous FTP website created with the WU-FTP server. In the following sections, we examine those measures.

Limiting Access

By default, all logins are directed to the `/var/ftp` directory. You can change that in `/etc/ftpaccess` by activating the following line for desired users:

```
# realuser user1,user2
```

If you remove the comment mark (`#`) and change *`user1`* and *`user2`* to real users on your system, the FTP server sends these users to their home directories when they log in—and they have access to higher-level directories such as root (`/`).

If you want all users to access your FTP server starting in the `/var/ftp` directory, comment out this line in `/etc/ftpaccess`.

Understanding the chroot Jail

The concept that protects other directories on an FTP server is the *chroot jail*. By definition, there is no higher directory than root (`/`). The `chroot` *`/abc/def`* command changes the effective root directory to *`/abc/def`*.

On an anonymous FTP server, the `/var/ftp` directory looks like the root (`/`) directory. The configuration for the anonymous FTP server applies the `chroot /var/ftp` command to all users who log into that server. If an anonymous user tries to run a command such as `cd /var` or `cd /etc`, it won't work, because higher-level directories are protected by the chroot jail.

Setting Up Command Limits

Access to dangerous commands can also be limited. By default, `/etc/ftpaccess` limits access to four commands, as shown. You may want to add other commands to the list. For example, if you make a command executable by an authorized user, you can add it to this list to prevent access by anonymous users.

```
chmod      no   guest,anonymous
delete     no   anonymous
overwrite  no   anonymous
rename     no   anonymous
```

Configuring Network File System Servers

The Network File System (NFS) is fundamental to Linux. In fact, one of the basic NFS configuration files is included in the same `setup-*` RPM package as `/etc/passwd` and `/etc/profile`. Yet managing NFS means that you need to pay attention to a number of different daemons.

Setting up exports from an NFS server is relatively easy. Basically, all you need to do is add a line for each shared directory to `/etc/exports` and share it with the network, and you're on your way. But pay attention to the syntax; the right commands help you secure the directories that you share through NFS.

One key to NFS is the remote procedure call (RPC), which allows you to seamlessly run commands on remotely mounted directories. All the NFS daemons use RPC.

The GUI configuration tool for NFS, `redhat-config-nfs`, can help you configure simple shared directories. Remember, this GUI tool is just a front end for what you'll learn in this chapter about configuring NFS.

NFS Packages

The packages you need for NFS may already be installed. Some of these packages are fundamental to a smoothly running Linux system. Table 22.5 describes the packages associated with NFS. As we explained in Chapter 10, you can run the `rpm -qi` *`packagename`* command to learn more about each package.

TABLE 22.5: NFS-RELATED RPM PACKAGES

PACKAGE	FUNCTION
`setup-*`	Shared NFS directories are defined in `/etc/exports`.
`initscripts-*`	Includes the basic scripts for mounting network directories during the boot process.
`nfs-utils-*`	Includes basic NFS commands and daemons.
`portmap-*`	Supports secure NFS remote procedure call (RPC) connections.
`quota-*`	Includes `rpc.rquotad` for quotas on directories shared over a network; this package is not required.

Basic Daemons

At least five Linux services are required to run NFS smoothly. They each relate to different functions, from mounting to making sure that remote commands get to the right place. These services are started through the `nfs`, `nfslock`, and `portmap` scripts in the `/etc/rc.d/init.d` directory. Here's a brief description of each daemon:

The basic NFS Naturally, there is an NFS server daemon, `rpc.nfsd`, that's started through the `nfs` script in the `/etc/rc.d/init.d` directory. The NFS daemon also starts the mount daemon (`rpc.mountd`) and exports shared directories. You can implement configuration changes by stopping and restarting the NFS service.

RPC mount While you can use the `mount` command to connect to local directories (such as from a floppy) or network directories (such as from a Samba server), there is a special daemon for mounting NFS directories: `rpc.mountd`.

The portmapper While the `portmap` daemon just directs RPC traffic, it is essential to NFS service. If `portmap` is not running, NFS clients can't find directories shared from NFS servers.

Reboots and *statd* There will be times when your connection to an NFS server goes down. You may have a scheduled reboot, or your server may just have crashed. The `rpc.statd` daemon works with `rpc.lockd` to help clients recover NFS connections after an NFS server reboots.

Locking When files are opened through a shared NFS directory, a lock is added. The lock prevents users from overwriting the same file with different changes. Locking is run through the `rpc.lockd` daemon, via the `nfslock` script.

Setting Up Exports

Shared NFS directories are listed in `/etc/exports`. As an example, assume you have one CD drive on your NFS server that you want to share with the other computers on your LAN. Normally, CDs are mounted on the `/mnt/cdrom` directory. You also want to share the `/tmp` directory to help share special packages. The format is simple:

```
sharedirectory    hosts(specs)
```

In other words, in `/etc/exports`, you specify the directory that you want to share, the computers that you want to share with, and the limits that you need. Let's look at a couple of examples of how you could do this.

```
/mnt/inst *.example.com(ro,sync) big.example.com(rw,sync)
/tmp  *(rw,insecure,sync,no_wdelay,anonuid=600)
```

The *shareddirectory* is self-explanatory. There are several ways to specify the computers that you want to share with; while you can use IP addresses, NFS does not recognize CIDR notation. Several examples are shown in Table 22.6.

TABLE 22.6: SPECIFYING HOSTS IN */ETC/EXPORTS*

EXAMPLE	EXPLANATION
`*.example.com`	All computers in the `example.com` domain
`newcomp`	The computer named newcomp
`10.11.12.13/255.255.255.0`	The network with the specified IP address and subnet mask

Finally, you must specify if and how you want to limit access to the shared directory. Do you want it shared as a read-only filesystem? Do you intend to share all subdirectories of a shared directory? Do you want to give the root user from a specific computer root-level access through the directory? While the options shown in this section may be a bit cryptic, you can specify these parameters and more in `/etc/exports`, as described in Table 22.7.

Table 22.7: */etc/exports* Shared Directory Specifications

Spec	Effect
`ro`	If a directory is mounted `ro`, users can have only read-only access to it (default).
`rw`	If a directory is mounted `rw`, users can read or write to it.
`sync`	All data is written to a share as requested.
`async`	NFS may respond to a request before writing data.
`secure`	NFS requests (default) are sent through a secure TCP/IP port below 1024; default medium- and high-security firewalls block these ports.
`insecure`	NFS requests are sent through TCP/IP ports above 1024.
`wdelay`	If more than one computer is about to write to a shared NFS directory, the writes are grouped together (default).
`no_wdelay`	If more than one computer is about to write to a shared NFS directory, the data is written immediately; if you've set `async`, this setting is not required.
`hide`	NFS by default shares directories, such as `/home/mj`, without sharing their subdirectories, such as `/home/mj/.kde`.
`no_hide`	When you share an NFS directory, this automatically also shares the subdirectories.
`subtree_check`	If you export a subdirectory such as `/usr/sbin`, this forces the NFS server to check lower-level directories (for example, `/usr`) for permissions (default).
`no_subtree_check`	If you export a subdirectory, such as `/home/mj`, it does not check the higher-level directory, such as `/home`, for permissions.
`insecure_locks`	For older NFS clients, this does not check if a user has read access to a requested file; same as `no_auth_nlm`.
`secure_locks`	For older NFS clients, this checks for user permissions on a requested file (default); same as `auth_nlm`.
`all_squash`	The UID and GID of exported files are mapped to the user anonymous; good for public directories.
`no_all_squash`	The UID and GID of exported files are retained (default).
`root_squash`	All requests from the user root are translated or mapped as if they came from the user anonymous (default).
`no_root_squash`	This allows the root user to have full administrative access through the shared directory.
`anonuid=xyz`	This specifies the UID of the anonymous user in the NFS server's `/etc/passwd` file.
`anongid=xyz`	This specifies the GID of the anonymous group in the NFS server's `/etc/group` file.

Starting with Red Hat Linux 8.0, you now need to specify `sync` or `async` for any shared NFS directory. In other words, you have to specify whether the shared directory responds to a command before a file is written permanently, such as to a hard disk.

Now that you've seen what can go into an /etc/exports file, return to the earlier example. It should make some sense to you now.

```
/mnt/inst *.example.com(ro,sync) big.example.com(rw,sync)
/tmp  *(rw,insecure,sync,no_wdelay,all_squash,anonuid=600)
```

The first line shares the /mnt/cdrom directory with all computers in the example.com domain. This directory is read-only, unless the connection is made from the computer named big.example.com. (Naturally, this works only if the media mounted on /mnt/inst, such as a CD, is writeable.)

The next line shares the /tmp directory with all computers. Computers that connect to this share can read or write (rw) to /tmp. The requests can be sent through TCP/IP ports above 1024 (insecure). Requests are written to /tmp before anything else is done (sync). Data is written immediately to disk, even if other computers that are sharing this directory are also about to write a file (no_wdelay). When mounting this directory, all users are given permissions associated with UID 600 in the NFS server's /etc/passwd file.

Securing NFS

NFS is inherently insecure. We recommend you limit access to shared NFS directories to computers inside your network. Allowing NFS connections through the Internet is strongly discouraged. As we've already shown, the commands associated with the /etc/exports file already add a layer of security. Shortly, we'll show you how to limit access to the portmap to a specific network, with appropriate commands in /etc/hosts.allow and/or /etc/hosts.deny.

NFS and an *iptables* Firewall

We discussed the basics of iptables in Chapter 17. As we discussed in that chapter, the Red Hat Enterprise Linux firewall tools redhat-config-securitylevel allow you to configure a standard firewall. You can also set the firewall to allow services through certain ports. Unfortunately, that's a problem for NFS, as four of the daemons (statd, mountd, lockd, and rquotad) may take random TCP/IP ports (above 1024).

There are two standard NFS related TCP/IP ports: 111 and 2049. As you can see from /etc/services, port 111 is related to the portmap daemon, and port 2049 is the channel for NFS.

To configure an iptables firewall for the other daemons, you'll need to tie them down. This process is documented online with the NFS HOWTO at www.tldp.org/HOWTO/NFS-HOWTO/index.html.

NFS and a TCP Wrappers Firewall

In Chapter 18, we discussed another Linux firewall related to xinetd services. With the wrong commands in /etc/hosts.deny, you can block the portmap, rpc.mountd, rquotad, statd, and lockd services. For example, the simplest firewall in /etc/hosts.deny blocks everything.

```
ALL:ALL
```

You may recall that xinetd reads /etc/hosts.allow first. So you can let the portmap through this firewall with a simple command. For example, you could add the following command to /etc/hosts.allow to let portmap through for the given network IP address (192.168.0.0):

```
portmap: 192.168.0.0/255.255.255.0
```

Use the same techniques with the other NFS-related services. Remember, CIDR notation such as 192.168.0.0/24 is not allowed in either the `/etc/hosts.allow` or `/etc/hosts.deny` file.

Starting NFS

You've configured exports. You've customized any firewall you may have. Finally, you're ready to start NFS and export the directories you plan to share.

Start with the `rpcinfo -p` command. If NFS is running properly, you should see entries for at least `portmap`, `nfs`, and `mountd`, similar to what is shown in Figure 22.8.

FIGURE 22.8
Checking NFS daemons

```
[root@Enterprise3d root]# rpcinfo -p
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  32768  status
    100024    1   tcp  32768  status
    391002    2   tcp  32769  sgi_fam
    100011    1   udp    791  rquotad
    100011    2   udp    791  rquotad
    100011    1   tcp    794  rquotad
    100011    2   tcp    794  rquotad
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100021    1   udp  32799  nlockmgr
    100021    3   udp  32799  nlockmgr
    100021    4   udp  32799  nlockmgr
    100021    1   tcp  33459  nlockmgr
    100021    3   tcp  33459  nlockmgr
    100021    4   tcp  33459  nlockmgr
    100005    1   udp    809  mountd
    100005    1   tcp    812  mountd
    100005    2   udp    809  mountd
    100005    2   tcp    812  mountd
    100005    3   udp    809  mountd
    100005    3   tcp    812  mountd
[root@Enterprise3d root]#
```

If you don't, NFS isn't ready, and you need to start these daemons. If necessary, you should be able to start the `rpc.mountd` and `nfs` daemons with the `service nfs start` command. You should also be able to start the `portmap` daemon with the `service portmap start` command.

Once the service is started, you can export the shared directories with the appropriate `exportfs` command. Some of the options are listed in Table 22.8.

TABLE 22.8: *EXPORTFS* COMMANDS

COMMAND	FUNCTION
`exportfs -a`	Exports all shared directories from `/etc/exports`
`exportfs -r`	Revises the list of shared directories after you've changed `/etc/exports`
`exportfs -u`	"Unexports" all directories
`exportfs -v`	Displays currently shared directories

Now you're ready to connect to a shared directory from an NFS client computer. But there's one thing left to do: Make sure that the right services will start the next time you boot Linux. As discussed

in Chapter 13, you can do this with the proper `chkconfig` command. The following commands check the runlevels at which the `nfs` and `portmap` daemons start:

```
# chkconfig --list nfs
# chkconfig --list portmap
```

And if necessary, the following commands make sure that these daemons start at the appropriate runlevels. When the `nfs` daemon starts, it also starts `rpc.mountd` and, if available, the `rpc.rquotad` daemon.

```
# chkconfig --level 235 portmap on
# chkconfig --level 235 nfs on
```

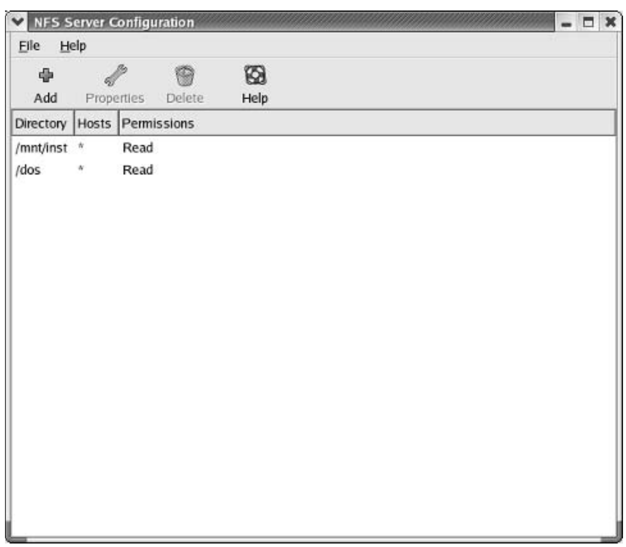
Configuring with *redhat-config-nfs*

You can also use `redhat-config-nfs` to configure your NFS server in a GUI. To start it, run this command, or in GNOME (or KDE), use the Main Menu ➢ System Settings ➢ Server Settings ➢ NFS command. This opens the NFS Server Configuration menu shown in Figure 22.9.

To start the configuration process, click the Add button. This opens the Add NFS Share window shown in Figure 22.10. We'll look at configuring the directories described earlier with `redhat-config-nfs`. For your reference, the previous commands from `/etc/exports` that we'll be emulating are:

```
/mnt/inst *.example.com(ro,sync) big.example.com(rw,sync)
/tmp  *(rw,insecure,sync,no_wdelay,all_squash,anonuid=600)
```

FIGURE 22.9
The NFS Server Configuration menu



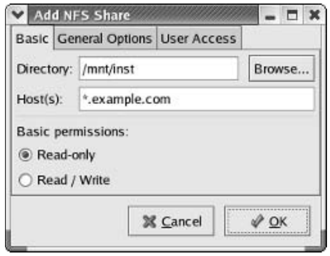


Figure 22.10
Adding a shared NFS directory

As shown in Figure 22.10, we've set up a share of the `/mnt/inst` directory, with read-only permissions for computers in the `*.example.com` domain. You can set up a separate share of `/mnt/inst` or `/tmp` to a specific computer such as `big.example.com` with read-write permissions.

Select the General Options tab, as shown in Figure 22.11. You can set up several of the options described in Table 22.9. By default, only Sync Write Operations On Request is active. Table 22.9 lists each option and its corresponding command.

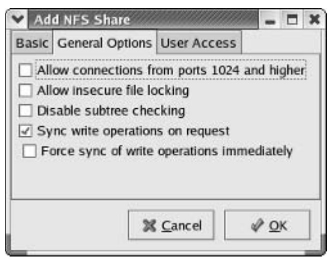


Figure 22.11
The General Options tab

Table 22.9: Add NFS Share General Options and Their Corresponding NFS */etc/exports* Commands

Option	NFS Command
Allow Connections From Ports 1024 And Higher	`insecure`
Allow Insecure File Locking	`insecure_locks`
Disable Subtree Checking	`no_subtree_check`
Sync Write Operations On Request	`sync`
Force Sync Of Write Operations Immediately	`no_wdelay`

Select the User Access tab, as shown in Figure 22.12. Table 22.10 lists each option and its corresponding command.

Figure 22.12
The User Access tab

Table 22.10: Add NFS Share User Access Options and the Corresponding NFS /*etc/exports* Commands

Option	NFS Command
Treat Remote Root User As Local Root	`no_root_squash`
Treat All Client Users As Anonymous Users	`all_squash`
Specify Local User ID For Anonymous Users; User ID	`anonuid=`*`userid`*
Specify Local Group ID For Anonymous Users; Group ID	`anongid=`*`groupid`*

You'll note that the first two commands on this tab are mutually exclusive; in other words, you can't treat a remote user as root if you've configured all NFS clients as anonymous users.

The Specify Local User ID and Specify Local Group ID options aren't configured with a corresponding NFS command; they make no sense and are therefore not activated unless you've set a specific user or group ID. For more information on user and group ID concepts, see Chapter 9.

The resulting `/etc/exports` file is slightly different from before; separate lines are required for the read-only and read/write setups to the computer and network specified earlier.

```
/mnt/inst *.example.com(ro,sync)
/mnt/inst big.example.com(rw,sync)
/tmp  *(rw,insecure,sync,no_wdelay,all_squash,anonuid=600)
```

Remember, you can't configure NFS with `redhat-config-nfs` alone; for example, you still need to make sure you don't have a firewall blocking NFS messages, as explained earlier in this chapter. You also should make sure that the `nfs` and `portmap` daemons are started at the appropriate runlevels the next time you boot Linux.

Working with NFS Clients

From a client computer, you need to know the available shared directories, the right way to mount these directories, and how to configure these directories to mount automatically the next time Linux boots on your computer.

Listing Shared Directories

It's easy to list the shared directories from an NFS server. All you need is the hostname or IP address of the server. The following command gives you the current list of shared directories from the computer named Enterprise3d:

```
# showmount -e Enterprise3d
Export list for Enterprise3d:
/tmp       *
/mnt/inst  *.example.com,big.example.com
```

TIP You can use the `showmount -e` *command on the NFS server to make sure it is actually exporting the directories you want to share through* `/etc/exports`*.*

This command lists two shared directories, along with the computers that are allowed to connect to each directory. If the list is not accurate, or the command does not work, take the following steps:

Check the daemons on the server. Are the `nfs`, `portmap`, and `mountd` daemons running?

Inspect the firewalls on both the server and the client. Do you have an `iptables` firewall in operation? If so, have you specified and allowed traffic through ports for the `lockd`, `statd`, `mountd`, and `rquotad` daemons? Are some of the NFS services blocked in `/etc/hosts.deny`?

Verify that you exported the directories in ***/etc/exports*****.** On the NFS server, run the `showmount -e` command. Remember, if you've just modified this file, you'll need to run the `exportfs -r` command on the server to refresh the export list.

If all else fails, refer to the discussion in Chapter 16 on network troubleshooting. While commands are available for testing network connectivity, the most common cause of network problems is the physical connection.

Mounting a Shared NFS Directory

Assuming everything is all right on the server, try the `showmount -e` *`NFSserver`* command again. Once you see the export list, you can mount one of these directories on your Linux computer. The following example mounts the `/tmp` directory from the computer named Enterprise3d on the local computer's `/tmp` directory:

```
# mount -t nfs Enterprise3d:/tmp /tmp
```

To translate, this mounts an NFS (`-t nfs`) filesystem, the `/tmp` directory from Enterprise3d on the local `/tmp` directory. But if there is a problem on the NFS server, or on the network connection, risks are involved. Suppose your connection "hangs," which locks up your console. Your computer will keep trying to connect, even if the NFS server has disappeared. Thus, a command with options (`-o`) such as the following is preferred:

```
# mount -t nfs -o soft,intr,timeo=50 Enterprise3d:/tmp /tmp
```

This adds options to soft mount, interruptible by the NFS server, with a timeout (`timeo`) of 50-tenths of a second (which of course corresponds to 5 seconds). But as discussed in Chapter 7, this command can be simplified. You can enter mount information in `/etc/fstab`:

```
Enterprise3d:/tmp    /tmp    nfs    soft,intr,timeo=50  0 0
```

Then all you would need to run to mount the shared `/tmp` directory is this command:

```
# mount /tmp
```

Summary

The File Transfer Protocol, FTP, is still in common use today. FTP is optimized for sharing files. Download speed for files is as important as ever. For example, you want to keep the time it takes to download a 650MB+ file for a Red Hat Installation CD to a minimum.

There are text-based and graphical FTP clients. While graphical clients such as gFTP are pretty, they are essentially front ends for the command-line FTP client. A substantial number of commands are available at the `ftp>` command prompt.

The default Red Hat FTP server is known as Very Secure FTP, or vsFTP. Its developers claim that it is more efficient than WU-FTP, the previous default Red Hat FTP server. The key configuration file is `/etc/vsftpd/vsftpd.conf`. In this file, you can configure anonymous access, messages, logging, uploading, and more.

You can also set up WU-FTP with real users, based on the user accounts in the FTP server's `/etc/passwd` file. Key configuration files in the `/etc` directory include `ftpaccess`, `ftpconversions`, and `ftphosts`. With the right changes, you can even configure user and time limits, as well as anonymous uploads on your FTP server. Several commands let you manage a WU-FTP server, including `ftpwho`, `ftpcount`, `ftpshut`, and `ftprestart`.

It's common to configure an anonymous FTP server. For this purpose, you also need the `anonftp-*` RPM package, which configures anonymous directories in `/var/ftp`. This works with vsFTP and WU-FTP. Anonymous users can't go above this directory because of the concept of the chroot jail. vsFTP can be configured for anonymous access. In WU-FTP, it's fairly easy to restrict access to anonymous users and critical commands in `/etc/ftpaccess`.

If you're sharing files between Linux and Unix computers, the standard service is the Network File System (NFS). Running NFS requires starting several `/etc/rc.d/init.d` scripts, including `nfs`, `nfslock`, and `portmap`. NFS directories are shared through `/etc/exports` and posted with the `exportfs` command. NFS communication can be blocked through `iptables` firewalls as well as TCP Wrappers rules in `/etc/hosts.allow` and `/etc/hosts.deny`.

Once you've shared a directory through NFS, you can mount it from an NFS client computer. The `showmount -e` *`NFSserver`* command lists shared directories. You mount an NFS server just like any other local or remote directory. Be sure to configure the mount in `/etc/fstab` in ways that do not "hang" when the NFS server is not available.

In the next chapter, we'll explore the authentication services for a network of Linux and Unix computers. The Network Information System (NIS) and the Lightweight Directory Access Protocol (LDAP) services allow you to configure a single database of login and other configuration files for a network.

Chapter 23

Linux Authentication Services: NIS and LDAP

ON A NETWORK WITH Linux and Unix computers, the two common authentication services are the Network Information Service (NIS) and the Lightweight Directory Access Protocol (LDAP). Both NIS and LDAP allow you to keep a common database of key configuration files on your network.

Every Linux computer normally has its own basic configuration files for users, such as `/etc/passwd` and `/etc/group`. On many LANs, it would be easier to configure all users with the same username and password. Without NIS or LDAP, that means making sure all users have an account on each computer—and each account has the same UID and GID numbers. This can be a cumbersome process. With NIS or LDAP, you can configure a single database of usernames, passwords, and a number of other configuration files. This chapter covers the following topics:

- Setting up Network Information Service servers
- Using NIS clients
- Setting Up the Lightweight Directory Access Protocol
- Configuring LDAP clients
- Running the Red Hat Authorization Configuration tool

Setting Up Network Information Service Servers

The Network Information Service (NIS) is a distributed database service that uses one set of configuration files for multiple computers on a LAN.

All Red Hat Enterprise Linux computers can be installed with the same basic configuration files. An NIS database of configuration files may be easier to maintain instead of different versions of the same file on different computers.

For example, you may want your users to be able to enter the same username and password at each of your computers. One approach is to copy the `/etc/passwd` and `/etc/group` files to every computer

on your LAN. Alternatively, you can configure a central database on an NIS server. In NIS, these databases are also known as *maps*.

You may have a larger LAN and don't want to go to the trouble of creating a DNS server. While you could copy the `/etc/hosts` file to every computer on your LAN, this becomes more difficult as you add more computers to your LAN. Storing `/etc/hosts` on a central NIS server map is an excellent option.

NOTE *While creating a DNS server is not difficult, it does mean running another service. Many administrators try to keep their services to a minimum. One way to do this is by avoiding the use of DNS on a local network, relying on* `/etc/hosts` *shared via NIS.*

Red Hat Enterprise Linux 3 comes with the packages needed to support a regular NIS version 2.*x* server. In this chapter, you'll notice a couple of references to `nisplus`, which is based on the NIS version 3.*x* server (reportedly troubled by bugs).

In the following sections, we'll examine the required RPM packages, define the domain, define the files that we're going to share, start the NIS services, and generate the database maps that we'll need.

NOTE *The drawback to NIS is security. If you have an NIS server, it should be on a LAN behind a firewall. You should not have a firewall between NIS servers and clients. While there are a number of things you can do with firewalls to help secure NIS on a LAN, it may be more trouble than it's worth. A simple web search for the terms* NIS *and* security *reveals many thousands of websites and messages detailing various security issues. While NIS remains popular, we've included LDAP in this chapter for that reason.*

NIS Packages

Four basic RPM packages are associated with NIS, as shown in Table 23.1. Notice that we have also included the `portmap-*` RPM; it's also used by NFS (see Chapter 22). As described in Chapter 10, you can run the `rpm -qi` *`packagename`* command to learn more about each package. The NIS server is part of the Network Servers package group; the other packages are installed by default.

TABLE 23.1: NIS RPM PACKAGES

PACKAGE	FUNCTION
`portmap-*`	Supports secure NIS remote procedure call (RPC) connections.
`ypbind-*`	The NIS client package; it binds a client to a server.
`ypserv-*`	The NIS server package.
`yp-tools-*`	Includes basic NIS commands.

NOTE *We're assuming you've already configured NFS on your computer, per Chapter 22. If you haven't, you'll need to make sure that the* `portmap` *daemon is running now and the next time you start Linux with the appropriate* `service` *and* `chkconfig` *commands.*

While the NIS server is governed by the `/etc/ypserv.conf` configuration file, we do not describe the file in any detail here, as the default configuration for Red Hat Enterprise Linux 3 is sufficient for most NIS configurations. For your reference, here are the commands associated with that file:

```
dns: no
files: 30
xfr_check_port: yes
* : * : shadow.byname : port
* : * : passwd.adjunct.byname : port
```

Briefly, with these commands, the DNS line is ignored, the `files` command supports caching, requests are supported only below TCP/IP port 1024, and the final two lines support shadow password mapping.

NOTE *NIS packages, and a number of NIS commands, start with* `yp`*, as it was originally known as the Yellow Pages. However, British Telecom owns a trademark for the Yellow Pages, which forced Sun Microsystems (the developer of NIS) to change the name.*

Defining the NIS Domain

NIS clients and servers are organized in domains. Unfortunately, NIS domains are unrelated to the domains associated with computer names, such as `linux.mommabears.com`, or even the domains associated with Microsoft networks.

First, your computer may already have an NIS domain name. To find out, run the `domainname` command. If it returns "(none)," your computer does not have an assigned NIS domain.

Assigning an NIS domain name is easy. For example, the following command defines a domain name of nistest for the local computer:

```
# domainname nistest
```

You'll also want to add a corresponding entry in `/etc/sysconfig/network` so it is known the next time you boot Linux. In this case, here's the line you need to add:

```
NISDOMAIN=nistest
```

You'll need to add this line to the `/etc/sysconfig/network` file on each NIS server and client on your network.

Defining Shared Files

Once you've installed the required RPMs and set the NIS domain name, the next step is to configure the NIS server. This starts with the `Makefile` in the `/var/yp` directory. It is an extensive file; essentially, you get to set parameters, and the script at the bottom processes the files you select into NIS database maps to be shared on the NIS domain. The variables we'll describe in this section are based on the default `Makefile` and are limited to that file.

You can configure NIS to look for computers that aren't in the NIS database. If you enable this command (by removing the #), it looks to your DNS servers for more information:

```
#B=-b
```

On larger LANs, you may have one or more backup (also known as *slave*) NIS servers. If you do, you'll want to change `true` to `false`. NIS then looks for the names of slave servers in `/var/yp/ypservers`.

```
NOPUSH=true
```

By default, Red Hat assigns user IDs and group IDs of 500 and above to regular users. Lower ID numbers, especially below 100, are normally associated with administrative and service users. The following commands exclude lower ID numbers from the appropriate NIS map database:

```
MINUID=500
MINGID=500
```

TIP *If you want to keep some "local-only" users, you can set higher ID numbers. For example, if you set* `MINUID` *and* `MINGID` *to 505, the first five users on each computer in the NIS domain will be local.*

If you try to connect to an NFS server on an NIS domain as a root user, the following commands map the root user ID to a special user known as nobody, which has few privileges:

```
NFSNOBODYUID=65534
NFSNOBODYGID=65534
```

As you may remember from Chapter 9, passwords are normally kept in `/etc/shadow` and `/etc/gshadow`. If your Linux system is configured this way, these commands incorporate passwords into the NIS map database:

```
MERGE_PASSWD=true
MERGE_GROUP=true
```

The following source directories should be standard. Unless you've changed the location of basic files such as `/etc/passwd`, you should not have to change any of these settings:

```
YPSRCDIR = /etc
YPPWDDIR = /etc
YPBINDDIR = /usr/lib/yp
YPSBINDDIR = /usr/sbin
YPDIR = /var/yp
YPMAPDIR = $(YPDIR)/$(DOMAIN)
```

Many of the following settings are standard. For example, `GROUP` is associated with `YPPWDDIR`, which is the `/etc` directory. From the first line in this list, the group configuration file is `/etc/group`, which is the standard location. If you've changed the location of any of these configuration files, revise these lines accordingly:

```
GROUP      = $(YPPWDDIR)/group
PASSWD     = $(YPPWDDIR)/passwd
SHADOW     = $(YPPWDDIR)/shadow
GSHADOW    = $(YPPWDDIR)/gshadow
ADJUNCT    = $(YPPWDDIR)/passwd.adjunct
```

```
#ALIASES   = $(YPSRCDIR)/aliases  # could be in /etc/mail or /etc/postfix
ALIASES    = /etc/aliases
ETHERS     = $(YPSRCDIR)/ethers
BOOTPARAMS = $(YPSRCDIR)/bootparams
HOSTS      = $(YPSRCDIR)/hosts
NETWORKS   = $(YPSRCDIR)/networks
PRINTCAP   = $(YPSRCDIR)/printcap
PROTOCOLS  = $(YPSRCDIR)/protocols
PUBLICKEYS = $(YPSRCDIR)/publickey
RPC        = $(YPSRCDIR)/rpc
SERVICES   = $(YPSRCDIR)/services
NETGROUP   = $(YPSRCDIR)/netgroup
NETID      = $(YPSRCDIR)/netid
AMD_HOME   = $(YPSRCDIR)/amd.home
AUTO_MASTER= $(YPSRCDIR)/auto.master
AUTO_HOME  = $(YPSRCDIR)/auto.home
AUTO_LOCAL = $(YPSRCDIR)/auto.local
TIMEZONE   = $(YPSRCDIR)/timezone
LOCALE     = $(YPSRCDIR)/locale
NETMASKS   = $(YPSRCDIR)/netmasks
YPSERVERS  = $(YPDIR)/ypservers
```

The `target: Makefile` command processes your NIS server, based on your NIS `domainname`. Generally, there is no need to change this command.

Next, you can select the files to share through your NIS server. The following list is from the default configuration file; you can add or subtract from the list by placing it in or removing it from the comment area (with the #):

```
all: passwd group hosts rpc services netid protocols mail \
   # netgrp shadow publickey networks ethers bootparams \
   # printcap amd.home auto.master auto.home auto.local \
   # passwd.adjunct timezone locale netmasks
```

The rest of the `Makefile` processes these settings. Because this is not a programming book, I won't cover additional configuration options; the default script in the rest of this file is sufficient for most users.

Creating a Database Map

Once you've configured the `/var/yp/Makefile`, the next step is to start the NIS server. Since it's a standard Linux service, simply issue the following command:

```
# service ypserv start
```

NOTE *The* `ypserv` *daemon won't work if you haven't defined an NIS domain name. As described earlier, you can do this with the* `domainname` ***`yourNISdomain`*** *command.*

Now you can process the `Makefile` into a database map. The `/usr/lib/yp/ypinit -m` command processes the `Makefile` to a `/var/yp/`*`dommainname`* subdirectory, where *`domainname`* is the name of the NIS

domain. You may realize that the `/usr/lib/yp` directory is not a part of the `PATH` (see Chapter 8), so you'll need to run the `ypinit` command using the full directory path.

NOTE *Don't forget to make sure that the NIS service starts the next time you boot Linux. The* `chkconfig --level 345 ypserv on` *command ensures that the NIS server starts automatically at runlevels 3, 4, and 5.*

When you run this command, you'll be prompted to enter the names of the computers that you want to add to your NIS domain. In the case shown, Enterprise3 is the name of the NIS server computer; you may have a computer with a name such as `linux.example.com`. The computers you add are included in `/var/yp/ypservers`, which means you can configure them as NIS slave servers.

```
# /usr/lib/yp/ypinit -m

At this point, we have to construct a list of the hosts which will run NIS servers.
Enterprise3 is in the list of NIS server hosts. Please continue to add the names
for the other hosts, one per line. When you are done with the list, type a <control
D>
          next host to add: Enterprise3d
          next host to add: Enterprise3m
          next host to add:
```

When you've finished adding computers to your NIS domain, you're asked to confirm your list. If you type **`n`**, you're prompted to start your list again. Otherwise, the `ypinit` command should start processing your `Makefile` with messages similar to those shown in Figure 23.1.

TIP *If you see an error starting with* `failed to send 'clear' to local ypserv`*, you probably forgot to start the NIS server, the* `ypserv` *daemon.*

```
The current list of NIS servers looks like this:

Enterprise3
Enterprise3d
Enterprise3m

Is this correct?  [y/n: y]  y
We need a few minutes to build the databases...
Building /var/yp/NISdomain/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/NISdomain'
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
Updating protocols.bynumber...
Updating protocols.byname...
Updating mail.aliases...
gmake[1]: Leaving directory `/var/yp/NISdomain'

Enterprise3 has been set up as a NIS master server.

Now you can run ypinit -s Enterprise3 on all slave server.
[root@Enterprise3 yp]#
```

FIGURE 23.1
Processing the NIS database

Updating the Database Map

If you need to update the NIS database, navigate to the `/var/yp` directory, and then run the `make` command. As you may remember from Chapter 12, a Linux `Makefile` can be typically processed in this fashion for a kernel, for many packages, and, yes, for the NIS database.

NIS Server Configuration Files

You'll need to do three things on each NIS server computer. Revise the `/etc/yp.conf` configuration file to point to the server. Add the name of the NIS domain to the `/etc/sysconfig/network` configuration file. And make sure the appropriate services are running and will start the next time you boot Linux.

The Default NIS Server Configuration: */etc/yp.conf*

It doesn't matter whether the computer is an NIS server or a client. Revising the `/etc/yp.conf` configuration file is simple. You need one command, in the following format:

```
domain NISdomainname NISservername NISserverIPaddress
```

In this line, the *`NISdomainname`* is the name of your NIS domain, which you can assign and check with the `domainname` command. The *`NISservername`* is the host name of the computer with your NIS server. And the *`NISserverIPaddress`* is straightforward; it isn't absolutely necessary as long as your NIS server IP address is assigned in `/etc/hosts`.

Including NIS In the Start Process: */etc/sysconfig/network*

My version of this file is straightforward; it tells Linux to start networking the next time my computer boots; it assigns the hostname, and it assigns the NIS domain name:

```
NETWORKING=yes
HOSTNAME=Enterprise3
NISDOMAIN=NISdomain
```

Starting NIS Server Services

It's easy to start NIS server services with the appropriate `service` and `chkconfig` commands. You just need to remember to start all the NIS server services.

`ypserv`	The NIS server.
`ypbind`	The NIS client.
`yppasswdd`	Transfers the common password database. (Don't forget the extra *d* at the end of `yppasswdd`.)
`ypxfrd`	Transfers NIS databases to slave servers.

Make sure to run the appropriate `service` and `chkconfig` commands to start these services, and make sure they start the next time you boot Linux.

Tip *If you configure an NIS master server, you may also want to configure an NIS slave server; then either server can respond to NIS broadcast requests.*

NIS Slave Servers

In larger networks, it's useful to have backups. The NIS slave server includes the information that the other computers on your network may need to keep going. To set up an NIS slave server, there are things you need to do on both the NIS master and the NIS slave computers.

Configuring the NIS Master

On the master, make sure you've added both computers to the list of NIS computers on the NIS domain, in `/var/yp/ypservers`. You should have already done this when you processed the `Makefile` with the `/usr/lib/yp/ypinit -m` command.

You also need to revise one line in the master NIS server's `/var/yp/Makefile` to show `NOPUSH=false`. This allows your NIS master server to copy its database to the NIS slave with the `yppush` command.

You'll also need to start the NIS map transfer server daemon, `ypxfrd`. Naturally, you can do this while Linux is running by using the `service ypxfrd start` command; to make sure it starts the next time you boot Linux, run the following command:

```
# chkconfig --level 345 ypxfrd on
```

Configuring the NIS Slave

Once the NIS master server is ready, there are just a few things that you need to check on the NIS slave. First, NIS slave servers should also be clients of both servers. For more information, read about NIS clients in the next section.

Go to the computer that you intend to set up as an NIS slave server. Make sure that computer is bound to the NIS master server. As long as you've assigned the NIS domain name on the slave computer, the `ypbind` command should do this automatically. The `ypserv` daemon should be running; you can check this with the `service ypserv status` command. Start these daemons as required. When you're ready, try the following command (substitute the hostname of your NIS master server for Enterprise3):

```
# /usr/lib/yp/ypinit -s Enterprise3
```

If the command is successful, you'll see a long series of messages, each of which transfers a configuration file from the NIS master to the NIS slave. One example is:

```
Transferring passwd.byname…
Trying ypxfrd … success
```

If you need to troubleshoot, you should see some messages here (and from a `ypbind -debug` command). Besides checking the network, recheck the NIS master server configuration process. Also, check that you've set `NOPUSH=false` on the NIS master to accommodate the NIS slave server. Make sure the appropriate services are started on the local NIS slave computer.

Note *By default, NIS does not use DNS servers, so it's important to have at least the NIS master server information in the NIS slave computer's* `/etc/hosts` *file.*

Using NIS Clients

Configuring your computer as an NIS client is easy; all you need to do is edit `/etc/yp.conf` and run the `ypbind` command. If you want to set up the computer as a permanent NIS client, just remember to run the `chkconfig --level 345 ypbind on` command to make sure that it starts at the appropriate runlevels.

There are a number of "`yp`" based commands that can help you test your connection. To make sure your NIS client computer actually uses some of the database map files, you must configure the `/etc/nsswitch.conf` file.

NIS Client Configuration in *yp.conf*

It's easy to configure an NIS client. Open `/etc/yp.conf` in a text editor. You'll see three basic commands:

```
domain NISDOMAIN server HOSTNAME
domain NISDOMAIN broadcast
ypserver HOSTNAME
```

The entries here are straightforward. Substitute the name of your NIS domain for *`NISDOMAIN`*. Substitute the name of the computer with the NIS server for *`HOSTNAME`*. If you also have a slave server, add the following command:

```
domain NISDOMAIN server NISSLAVEHOSTNAME
```

where *`NISSLAVEHOSTNAME`* is the hostname of the NIS slave server. Now you're ready to start the NIS client with the `service ypbind start` command.

TIP *If* `ypbind` *is having problems communicating with the NIS server, check for a firewall on the NIS server—it may be blocking NIS communication. Generally, NIS should be run on a LAN protected only from outside networks by a firewall.*

NIS Client Commands

There are a number of commands that you can use as an NIS client. Conveniently, they all start with the letters `yp`. They enable you to set passwords on the remote NIS server database, test the connection, read files from the NIS server database, and more. We take a look at these commands in the following sections.

YPCAT

The `ypcat` command reads files available from an NIS server database. Like the regular `cat` command, it just scrolls the information available from the file. However, what you see in an NIS client may vary slightly from the actual file on the server. For example, the following command just lists the `/etc/passwd` information for users with an UID >= 500 (unless you've changed the `MINUID` and `MINGID` variables in `/var/yp/Makefile`):

```
# ypcat passwd
```

YPCHFN

The `ypchfn` command changes the finger information on the NIS server database map. Like the `chfn` command, it normally applies to the current user. If you're in the root account, you run the `chfn` *`username`* command to change the finger information for the user of your choice.

As described in Chapter 18, you can store finger information, such as a user's full name and telephone number, in the fifth field of that user's entry in `/etc/passwd`.

Thus, the following command prompts you to change the finger information for user mj on the Enterprise3 NIS server. It also provides a series of prompts to help you revise user mj's finger information.

```
# ypchfn mj
Changing NIS account information for mj on Enterprise3.
Please enter root password:

Changing full name for mj on Enterprise3.
To accept the default, simply press return. To enter an empty field, type the word
"none".
Name [Michael Jang]:
```

YPCHSH

The `ypchsh` *`username`* command changes the default shell for a specific user in the NIS server's `/etc/passwd` file. It works in a similar way to `ypchfn`; this command prompts you for the NIS server root password and then prompts you to change the shell.

YPMATCH

The `ypmatch` *`username`* `passwd` command is an easy way to search through the NIS database file for your LAN's username entry in the master NIS server's `/etc/passwd` file.

YPPASSWD

The `yppasswd` *`username`* command allows you to change the password for a user on the NIS server. The user will have to use the new password to log onto any NIS client computers. Like the `ypchfn` and `ypchsh` commands, you're prompted for the NIS server root password before you're prompted to enter the new password for the desired user.

Configuring */etc/nsswitch.conf*

If you have an NIS server on your network, you'll want to make sure that the `/etc/nsswitch.conf` file on the NIS client looks for an NIS server for any associated configuration files. It also can point your client computer to other sources, such as the local configuration files.

For example, if you don't have an NIS server, your `/etc/nsswitch.conf` should be simple, with commands such as these:

```
passwd:  files
shadow:  files
group:   files
hosts:   files dns
```

Each of these commands specifies a search order. For example, the `hosts` line specifies a search through the local file (`/etc/hosts`) before moving onto a DNS server (which matches the configuration in `/etc/host.conf`). However, if you have an NIS server, you should include it in the list. For example, the following lines look to a properly bound NIS server database first:

```
passwd:  nis files
shadow:  nis files
group:   nisfiles
```

The `nis` entry corresponds to the standard NIS server. If you're using NIS version 3.*x*, you'll want to replace that entry with `nisplus`.

If you want to use the central NIS server `/etc/hosts` database, add a corresponding entry in `/etc/host.conf`. For example, the following directs your computer to first search through the NIS `/etc/hosts` database, then search the local `/etc/hosts`, and then finally search any DNS servers in `/etc/resolv.conf`:

```
order nis,hosts,bind
```

Setting Up the Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol (LDAP) essentially sets up a "white pages" type of service for the users and computers on a network. It is literally a lighter-weight version of the older X.500 protocol, associated with commands such as `whois` and `finger`. Developed at the University of Michigan, it allows you to organize the information for groups in one central database. As it supports the Secure Sockets Layer (SSL) and Transport Layer Security Protocols (TLS), it is more secure than services such as NIS.

LDAP is rich and complex directory database and authentication software. This is just a very basic introduction. The open-source implementation of LDAP, naturally, is known as OpenLDAP. More information on this project is available at `www.openldap.org`. The Red Hat Enterprise Linux OpenLDAP RPMs start with `openldap-*`.

Installing OpenLDAP Packages

There are four basic RPMs associated with the OpenLDAP service. Three are fairly self-explanatory based on the names: `openldap`, `openldap-server`, and `openldap-client`. The fourth, `nss_ldap`, supports password authentication and Pluggable Authentication Module (PAM) security. (For more information on PAM, see Chapter 17.) While the names are fairly self-explanatory, they are distributed among the Network Servers and System Tools package groups. So you can choose to install them directly with the appropriate `rpm` command.

Many of the commands associated with the LDAP server start with `slapd*`; the commands associated with the LDAP client start with `ldap*`. When you configure and add data to the LDAP server, make sure to use the associated `slapd*` commands.

NOTE *Take care to use the appropriate* `slapd*` *commands when configuring an LDAP client and server. Older versions of LDAP (and the associated online documentation) supported the use of* `ldap*` *commands to configure a server.*

Basic LDAP Definitions

LDAP definitions and configuration files include their own unique format and language. We therefore include a basic set of definitions in Table 23.2.

TABLE 23.2: LDAP DEFINITIONS

TERM	DESCRIPTION
Attributes	Other information associated with an entry; for example, a user's phone number is an attribute of that user.
`cn`	Common name.
`dc`	Domain component.
`dn`	Distinguished name.
entry	A unit within an LDAP directory.
`l`	Location.
LDIF	LDAP Data Interchange Format—an ASCII representation of an LDAP entry.
`mail`	E-mail address.
`o`	Organization name.
`objectClass`	Organization category.
`ou`	Organizational unit.
`rootdn`	Name of the user with no access controls on LDAP.
schema	Fundamental data structures associated with LDAP; stored in the `/etc/openldap/schema` directory.
`sn`	Surname.
`uid`	User ID.
`url`	Web page.

This is just a limited set of LDAP definitions. Additional terms are included in the LDAP Implementation HOWTO, available online at `www.tldp.org/HOWTO/LDAP-Implementation-HOWTO`.

Configuring an OpenLDAP Server

When you configure an OpenLDAP server, you'll need to configure the `/etc/openldap/sldap.conf` file. This configuration file defines how the LDAP server runs on your computer. Naturally, you may also want to configure your LDAP server as a client. We describe the process shortly. Next, we'll start the LDAP service. Then we'll add and check the needed entries in the LDAP database files.

/ETC/OPENLDAP/SLDAP.CONF

The default `sldap.conf` LDAP configuration file starts with a number of `include` directives, which carry the data structures defined in the `/etc/openldap/schema` directory to your LDAP server.

```
include     /etc/openldap/schema/core.schema
include     /etc/openldap/schema/cosine.schema
include     /etc/openldap/schema/inetorgperson.schema
include     /etc/openldap/schema/nis.schema
include     /etc/openldap/schema/redhat/autofs.schema
include     /etc/openldap/schema/redhat/kerberosobject.schema
```

These are standard LDAP data structures and should not be changed. If you're more comfortable with LDAP, you can configure your own data structure schema in the `/etc/openldap/schema/local.schema` file. If you choose to do so, you'll need to add one more `include` directive to this list.

```
include     /etc/openldap/schema/local.schema
```

The standard LDAP database is known as LDBM, as defined by the next active line in the default `sldap.conf` configuration file:

```
database     ldbm
```

Now you can define your LAN and associated attributes with a number of different `suffix` directives. The next line by default is:

```
suffix    "dc=my-domain,dc=com"
```

You'll want to change it to reflect your LAN domain name. If you're using `example.com`, you'll want to change it to:

```
suffix    "dc=example,dc=com"
```

As shown by the comments, you can add more `suffix` directives to reflect other attributes of your organization.

```
suffix    "o=Writers,c=US"
```

You also need to define the administrative user for the LDAP directory with the `rootdn` directive. You could change this to reflect the local root user and domain name.

```
rootdn     "cn=root,dc=example,dc=com"
```

You'll need to create a root password for your LDAP database and assign it to the `rootpw` directive. As the password may be transmitted over your network, you should encrypt this password. You can create an encrypted password for LDAP with the `slappasswd` command and transfer it to the `slapd.conf` file. The command may look like:

```
rootpw     {SSHA}TRhJAGOyWGJMFjn8+nW3on6Pjh5tJwR+
```

Once you've set up the LDAP database, you can comment out this line. It'll minimize the risk that someone will copy and decrypt your root LDAP password.

Transferring Encrypted Passwords

Sometimes you'll need to create and transfer encrypted passwords from the command line to a configuration file. You may need to do this for configuration files associated with a number of different software packages, such as those associated with an OpenLDAP server or the GRUB bootloader. The process is straightforward. In fact, if you're in a GUI such as GNOME, it's easy; all you need to do is highlight, right-click, and use the Copy and Paste commands from the pop-up menu.

However, the process is a little more difficult if you're working from the text console. You will need a mouse or other pointing device. To create an encrypted password for the `slapd.conf` configuration file, follow these steps:

1. Run the `slappasswd` command.
2. You'll be prompted to enter your desired LDAP root password twice.
3. The `slappasswd` command returns an encrypted version of your desired password. It'll start with `{SSHA}`, which tells us that these are Secure Hash passwords.
4. Highlight the password. With your mouse, it may be easier to start with the left side of the password.
5. Try a right-click. You should see the encrypted password repeated at the command line.
6. Open the `/etc/openldap/slapd.conf` file in a text editor. Input the `rootpw` directive. Right-click after the directive; you should see the password transferred here.
7. Save your changes to the `slapd.conf` file.

If you've installed the `openldap-servers` package, you should find the LDAP database directory, `/var/lib/ldap`. The permissions on that directory (700) make it accessible only to the ldap user who owns this directory. Finally, the following index directive specifies the information to be collected for each user:

```
index objectClass,uid,uidNumber,gidNumber,memberUid     eq
index cn,mail,surname,givenname                eq,subinitial
```

Make any changes required to accommodate your LAN, and save.

Starting LDAP

Once you've configured the LDAP server, the process of starting LDAP is straightforward. If you've read through this book, you should have seen variations of the following commands a number of times. This first command starts the LDAP server service (`slapd`):

```
# service ldap start
```

This second command makes sure that LDAP starts the next time you boot Linux into the standard runlevels.

```
# chkconfig --level 35 ldap on
```

This command actually starts `slapd`, the stand-alone LDAP daemon.

Adding Data to an LDAP Server Database

Now you can set up the LDAP database for your organization. You'll want to configure a database file in LDAP Data Interchange Format (LDIF). For my own personal network, I've added the following commands to a text file that I've named `example.ldif`.

```
dn: dc=example,dc=com
dc: example
o: Writers
objectClass: organization
objectClass: dcObject

dn: cn=root,dc=example,dc=com
cn: root
sn: Jang
objectClass: person
```

We've already defined these variables in Table 23.2. When you set up your own LDIF file, you need to be careful about the following:

- The definitions in the LDIF file must be consistent with the LDAP server configuration file, `/etc/openldap/slapd.conf`.
- Take care to avoid white space in the LDIF file. Extra spaces at the end of lines may result in errors when you try to set up your LDAP server.

We add this information to our LDAP server database with the following command:

```
# slapadd -l example.ldif -v -b "cn=root, o=Writers, c=us"
```

This command reads from the given LDIF file (-l), returns messages in verbose mode (-v), and specifies a suffix (-b). The suffix defines your organization, in case you have more than one LDAP database. If your database addition is successful, you'll see messages such as the following:

```
added: "dc=example,dc=com" (00000004)
added: "cn=root,dc=example,dc=com" (00000005)
```

Now you can repeat this process for other users. For example, if you wanted to add user elizabeth to this database, you could set up another LDIF file with the following information:

```
dn: cn=elizabeth,dc=example,dc=com
cn: elizabeth
sn: Zinkann
objectClass: person
```

Migrating Authentication Data to LDAP

You can copy a lot of data from your computer's basic configuration files to an LDAP server. To do so, you can use one of the migration scripts located in the `/usr/share/openldap/migration` directory.

These scripts are written in the Perl programming language, which should be installed with the Development Tools package group. As you can see in Figure 23.2, the list of scripts is extensive.

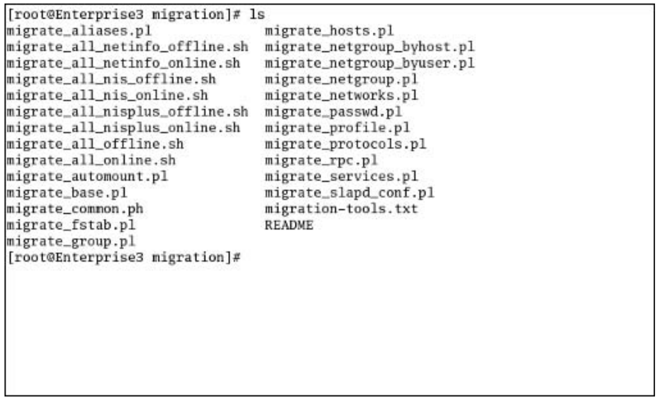


FIGURE 23.2
Default migration scripts

Before these scripts actually migrate data to your LDAP server, you need to modify two directives in the `migrate_common.ph` script. I've modified these directives as follows:

```
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = "example.com";
# Default base
$DEFAULT_BASE = "dc=example,dc=com";
```

Now we can use the scripts in this directory to migrate the data from our basic configuration files. For example, we can use the following commands to set up LDIF files from the basic password and group authentication files:

```
# /usr/share/openldap/migration/migrate_passwd.pl /etc/passwd passwd.ldif
# /usr/share/openldap/migration/migrate_group.pl /etc/group group.ldif
```

Then you can use the `slapadd` command to add this information to your LDAP server database.

```
# slapadd -l passwd.ldif -v -b "cn=root,o=Writers,c=us"
# slapadd -l group.ldif -v -b "cn=root,o=Writers,c=us"
```

Now your LDAP server is ready, and you can configure the local computer, as well as others on the local network, as LDAP clients.

Configuring LDAP Clients

There are two ways to configure a LDAP client. You can modify the appropriate configuration files, `/etc/ldap.conf` and `/etc/nsswitch.conf`. Alternatively, you can use the Red Hat Authentication Configuration tool to modify these tools using a graphical interface.

To configure an LDAP client that reads a remote database for usernames and passwords, you have to modify several PAM configuration files in `/etc/pam.d`. The process is long and somewhat risky. If you make a mistake, it may keep you from logging into your Linux computer. If you haven't properly backed up your `/etc/pam.d` configuration files, you may have big problems.

If you're willing to take these risks, there are a series of substitute PAM configuration files available in the `/usr/share/doc/nss_ldap-207/pam.d` directory. A detailed analysis of how each of these files works is beyond the scope of this book. For more information, see `www.padl.com`.

Configuring LDAP Clients in */etc/ldap.conf*

You can configure a wide variety of applications in `ldap.conf`, from sendmail to GNOMEMeeting. However, we'll just keep it simple in this book. All you need to do to configure your computer as an LDAP client is add four commands to this file:

The IP address of the LDAP server:

```
host 127.0.0.1
```

The name of the LDAP server search database:

```
base dc=example,dc=com
```

Current encryption; since we did not set up SSL or TLS encryption, the following command is appropriate:

```
ssl no
```

Finally, we need to reflect the current state of authentication. By default, Red Hat Enterprise Linux uses Pluggable Authentication Modules and MD5 encryption for passwords:

```
pam_password md5
```

Configuring */etc/nsswitch.conf*

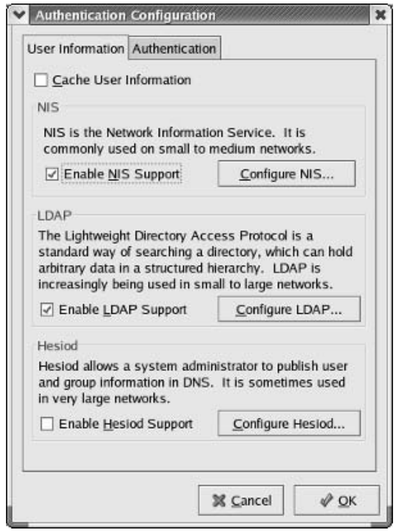
Finally, you'll want to configure the Name Server Switch configuration file, `/etc/nsswitch.conf`, to look to LDAP for authentication information. This is easily done with the following commands:

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
```

Running the Red Hat Authorization Configuration Tool

The Red Hat Authorization Configuration tool is a straightforward way to configure a connection to a running NIS or LDAP server. There are text and GUI versions of this tool, which you can start with the `redhat-config-authentication` command. For the purpose of this book, it's easier to illustrate the GUI version of this tool, as shown in Figure 23.3.

FIGURE 23.3
Authentication Configuration



This tool is fairly straightforward. If you want to configure this computer as an NIS client, you'll need to Enable NIS Support, as shown in the figure. You'll also need to click Configure NIS, where you can enter the name of the NIS domain as well as the name or IP address of the computer you've configured as an NIS server (master or slave) for your network.

If you want to configure this computer as an LDAP client, you can Enable LDAP Support under both the User Information and Authentication tabs. You'll need the LDAP Search Base DN, which we configured earlier as:

```
dc=example,dc=com
```

You'll also need the name or IP address of the LDAP server. Unless you have a reliable DNS service on your network, you should specify the IP address of that server.

This tool automatically makes the required changes to the appropriate configuration files, as described throughout this chapter.

Summary

The Network Information Service (NIS) shares configuration files with Linux and Unix computers. The Lightweight Directory Access protocol can do the same thing.

For example, you can use NIS to create a single database of usernames and passwords by converting an `/etc/passwd` and `/etc/groups` file on a server into a single shared database. You need to define an NIS domain name and shared files in `/var/yp/Makefile`. Once your `Makefile` is ready, you can convert it to a database with `ypinit`; changes can be processed with the `make` command in the `/var/yp` directory. Slave servers can also be configured with `ypinit` and refreshed with `yppush`.

Configuring an NIS client is relatively easy; just `ypbind` it to the appropriate server. Alternatively, you can use the `authconfig` command. Once you've connected, NIS client commands let you look

through the available databases. Finally, /etc/nsswitch.conf, properly configured, points your NIS client computer to the appropriate database on your NIS server.

Configuring a LDAP server requires the three openldap* and nss_ldap RPM packages. You can configure an LDAP server using the /etc/openldap/slapd.conf file. Once configured, you can start the LDAP server and then add organization and user data in LDIF format. You can also use scripts in the /usr/share/openldap/migration directory to transfer data from standard configuration files such as /etc/passwd to your LDAP server database. Be careful; unless the databases are consistent and you use the appropriate slap* commands, you may have trouble with this process.

When you configure an LDAP client, you need to modify the /etc/ldap.conf file to point to the appropriate LDAP server and the /etc/nsswitch.conf file to look to LDAP for basic information.

Finally, you can use the Red Hat Authentication Configuration tool to set up your Linux computers as a client to an NIS and or a LDAP server.

In the next chapter, we'll examine Samba, which allows you to share files and directories with Linux, Unix, and Microsoft Windows computers.

Chapter 24

Making Samba Work for You

With Samba, you can make your Linux computer a part of a Microsoft-based network. In this chapter, you'll learn how to configure Samba as a client and as a server on a network of Microsoft Windows computers.

Computers with various Microsoft operating systems can communicate with each other using the Server Message Block (SMB) protocol. When a Microsoft operating system shares files or printers on a TCP/IP network, it uses the Common Internet File System (CIFS). Samba is the way a Linux computer communicates with SMB and CIFS.

Samba is a heterogeneous service. Once you've configured Samba, other Microsoft Windows computers won't be able to tell the difference. Like CUPS from Chapter 20, Samba includes its own web-based configuration utility, SWAT, the Samba Web Administration Tool.

You can use Samba packages to configure your Linux computer as a server or a client and then connect to or share directories and printers. As a Samba client, you can also connect to a shared Microsoft directory in a terminal mode that looks like a text-based FTP connection.

The main Samba configuration file is `/etc/samba/smb.conf`. Many Linux administrators configure it directly in a text editor, and you can learn how to do the same to share directories and printers from your Linux computer. It's easy to test and troubleshoot the changes you make to `smb.conf`.

You can configure Samba accounts. If the Windows and Linux user names are the same, you can configure the associated accounts in `/etc/samba/smbpasswd`. If the usernames on a user's Linux and Windows accounts are different, you can set up a relationship in `/etc/samba/smbusers`. If you're setting up a Microsoft-style domain, you can also configure the required computer accounts on a Samba server.

Red Hat has included its own GUI configuration tool for Samba, known as the Red Hat Samba Server Configuration tool. While Red Hat does not include the distribution-neutral SWAT in the installation CDs, this package is available through the Red Hat Network. Red Hat's Samba Server Configuration tool can help you configure basic settings for your Samba server and shared directories. This chapter covers the following topics:

- Bridging the gap between Linux and Microsoft Windows
- Configuring Samba as a client

- Understanding the Samba configuration file
- Managing Samba users and computers
- Using SWAT
- Using the Red Hat Samba Server Configuration tool

Bridging the Gap between Linux and Microsoft Windows

As a heterogeneous service, Samba bridges the gap between Linux and Microsoft Windows—which essentially means it can communicate equally well with either operating system. In fact, you can configure Samba to share directories and printers in the same way as any other member of a Microsoft Windows network.

Functioning on a Microsoft Network

One of the advantages of Samba is that it allows you to configure a Linux or Unix computer to function in different ways on a Microsoft Windows network. When your configuration is complete, Microsoft users don't even need to know they're communicating with a Linux computer. With Samba, you can configure your Linux computer to look like any of the following types of computers:

- Member of a Microsoft Windows workgroup
- Member of a Microsoft Windows domain
- Microsoft Windows member server (even on a Windows 2000/2003 Active Directory network)
- Microsoft primary domain controller (PDC)

NOTE *While Samba does not explicitly allow you to configure Linux as a BDC, it is possible ; see the Samba BDC HOWTO at* `www.samba.org` *for more information.*

Samba was originally based on Microsoft's LAN Manager system, where client computers used NetBIOS names over the TCP/IP network, NBT (NetBIOS over TCP/IP); it does not need Microsoft's other networking system, NetBEUI. For more information on NetBIOS and NetBEUI, see Chapter 15.

Licensing

Don't let the title of this section make you panic. Samba is licensed under the GPL and is freely available as a part of different Unix-style operating systems, including Red Hat Enterprise Linux 3.

Samba makes it possible for you to set up Linux computers as part of a Microsoft network. It can reduce the number of Microsoft operating systems that you need to purchase for your network. As of this writing, you don't need to pay for any Microsoft license to use Samba.

NOTE *There may be some delays in compatibility between Samba and the next Microsoft operating system release (code named Longhorn). Current Microsoft plans include a new file system (WinFS) which reportedly is not compatible with current versions of Samba. Microsoft has taken a number of patents related to WinFS. However, people in the open-source community are, by definition, skilled at "reverse engineering." I have no doubts that Linux with Samba will be able to work with the Windows Longhorn WinFS soon after its release (our estimate is around 2006).*

Definitions

This chapter contains a few terms that are either exclusive to Samba or more closely related to the world of Microsoft networking. They include:

Primary domain controller (PDC) The computer that has the central database of usernames and passwords. It often also contains the central database of Microsoft Windows logon profiles.

Backup domain controller (BDC) This computer gets its information from a PDC. PDC and BDC are Windows NT concepts.

Browse list A list of shared resources on a network.

Active Directory The directory service associated with Windows 2000/2003.

Browse master A computer in charge of maintaining a browse list for a network.

Domain A network with a centralized database of at least usernames and passwords. This concept is quite different from an Internet domain name.

Member server Any computer on a Microsoft Windows network that shares directories or printers and is not a PDC or a BDC.

Peer-to-peer A group of computers on a LAN, each of which can act as a server; commonly associated with a workgroup.

Server A computer that shares directories or printers.

Share Any directory or printer that is shared on a network.

Workgroup A LAN without a dedicated server. Each computer is responsible for its own usernames and passwords; each computer often shares directories and printers with the rest of the LAN.

NOTE *In this chapter, the term* Microsoft server *on a network can refer to any Microsoft operating system that shares directories or printers. It can also refer to a Samba server on a Linux computer.*

Packages

Five basic packages are associated with Samba on Red Hat Enterprise Linux. All you need to configure your computer as a Microsoft client is `samba-client-*` and `samba-common-*`. The other packages help you configure your computer as a server on a Microsoft-style network. These packages are summarized in Table 24.1

Table 24.1: Samba RPM Packages

Package	Description
samba-*	The basic Samba server package, this includes commands for matching Linux and Microsoft usernames and passwords.
samba-client-*	This package allows you to set up your Linux computer to read shared Microsoft directories and print to shared Microsoft printers.
samba-common-*	This package includes files required to support Linux as a Samba client and as a Samba server.
samba-swat-*	This GUI tool lets you modify the main Samba configuration files, especially `smb.conf`; if you don't need fine-grained control, you may consider `redhat-config-samba` as an alternative. This is not available on the Red Hat Enterprise Linux CDs, but it can be downloaded through the Red Hat Network and the "rebuild" sites.
redhat-config-samba-*	This is the alternative to `samba-swat`; it's simpler but less mature and allows less configuration control.

Tip *While Red Hat does not include* `samba-swat` *in its Enterprise CDs, it is available as of this writing as an individual download through the Red Hat Network enterprise channel. This requires an official subscription to the Red Hat Network. If you're using one of the "rebuilds," you may need to search different directories on the respective download servers to find the* `samba-swat` *RPM, such as Extras or Addons.*

Configuring Samba as a Client

With the `samba-client-*` and `samba-common-*` RPM packages, you can see the directories and printers shared from a Microsoft computer. You can connect to a shared directory in two basic ways: by mounting a shared Microsoft Windows directory on a local Linux directory and by connecting to the shared directory in terminal mode, as if you were connecting to an FTP server. In addition, you need to know how to connect to a shared printer connected to a Microsoft computer.

Shared Samba Directory

It is easy to connect to a shared directory from a Microsoft server. As shown in Figure 24.1, all you need is the `smbclient` command, along with the name or IP address of the server. The command is slightly unusual; note the backslashes associated with the `smbclient` command.

There are a number of interesting shares in this directory. If the appropriate permissions are set on the Microsoft server, you can mount any of these shared directories almost like an NFS directory. The difference is subtle. For example, if you're the root user, the following command can mount the `Downloads` directory shared from the computer named allaccess on the local directory `/root/downloads`. Linux follows up by requesting a password.

```
# mount '//allaccess/downloads' /root/downloads
Password:
```

FIGURE 24.1
Reviewing the shares from a Microsoft server

```
[root@Enterprise3 root]# smbclient -L \\allaccess
Password:
Domain=[ALLACCESS] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

        Sharename      Type      Comment
        ---------      ----      -------
        RHEL3          Disk      Mastering Red Hat Enterprise Linux Book
        IPC$           IPC       Remote IPC
        D$             Disk      Default share
        print$         Disk      Printer Drivers
        SharedDocs     Disk
        Hentzen        Disk      Linux Transfer for Windows
        RHCE4          Disk
        ftproot        Disk
        HPLaserJ       Printer   Comment Test
        Downloads      Disk
        RedHat         Disk
        ADMIN$         Disk      Remote Admin
        C$             Disk      Default share
        Auctions       Disk
        ds             Disk
        Isabel         Disk
        Printer        Printer   hp psc 1200 series
        L$             Disk      Default share
Domain=[ALLACCESS] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
[root@Enterprise3 root]#
```

The mount command actually serves as a "front end" to the `mount.smbfs` command, from the `samba-client` RPM package.

NOTE *Strictly speaking, you should specify the Samba file type for* `mount` *by using the* `-t smbfs` *switch; because* `mount` *is a "front end" for* `mount.smbfs`*, this is not required. This changes with Linux kernel 2.6, which includes support for a specific CIFS module that more closely matches the latest Microsoft filesystem.*

Since there is no username, this `mount` command will work only with a Microsoft Workgroup–style shared directory. In other words, this requires a shared directory from a Windows 95/98/ME computer—or a shared directory where the user *Everyone* is allowed access—and you need just the password associated with the shared directory (if required). (Microsoft Windows NT/2000/XP computers can also be configured in this manner.)

However, most networks are more restrictive. On Microsoft Windows servers, you can limit access to specific users and groups. In that case, you must have a username and password with appropriate privileges to that directory. The `-o` option allows you to enter usernames, passwords, and more when specifying a share. I personally prefer to specify just the username in the command line so I don't have to type the password in clear text in a terminal. Linux automatically prompts for a password.

```
# mount -o username=michael '//allaccess/downloads' /root/downloads
Password:
```

NOTE *You can even provide variable levels of access; for example, you can configure read-only access for guest users while providing full access to privileged users.*

You can go further; for example, if you want to specify a username from a PDC that is allowed to access a directory on a member server, you can set that as part of the username. For example, if the domain is GRATEFUL and the PDC username is mj, this command may look like:

```
# mount -o username=GRATEFUL/mj '//allaccess/downloads' /root/downloads
Password:
```

If you're acting as a regular user, you could also substitute the `smbmount` command. To set this up, you'll have to set superuser ID (SUID) permissions on the `smbmnt` command (yes, when you change permissions on `smbmnt`, regular users are allowed to use `smbmount`). You'll also want to allow regular users to unmount Samba directories. To do so, run the following commands:

```
# chmod u+s /usr/bin/smbmnt
# chmod u+s /usr/bin/smbumount
```

Then you can use the `smbmount` and `smbumount` commands as a regular user. For example, if you wanted to browse user donna's Windows My Documents on the Microsoft computer named allaccess, you'd connect with the following command:

```
$ smbmount "//allaccess/My Documents" /home/michael/shares -o
➥ username=donna
Password:
$
```

As a regular user, you'll see the `$` as a regular bash shell command prompt. To unmount this share, you'd run the following command:

```
$ smbumount /home/michael/shares
```

The specifications for a Windows server can get more complex. Some of the other available Samba `mount` options are shown in Table 24.2.

TABLE 24.2: SAMBA *MOUNT -O* OPTIONS

OPTION	DESCRIPTION
`username=`*winuser*	Allows you to specify the Microsoft username of an authorized user on the Microsoft server.
`password=`*winpass*	Lets you specify the Microsoft password associated with the privileged Microsoft user; if you enter only a username, you're automatically prompted for a password.
`credentials=`*file*	Reads a username and password from a specified file, which you can protect, such as `/etc/shadow`; useful for automatic mounting from files such as `/etc/fstab`. The syntax is: `username=`*winuser* `password=`*winpass*
`uid=`*linuser*	Allows you to set the Linux users who own the files on the mounted filesystem; can be a User ID number or a username.
`workgroup=`*winwork*	Lets you specify the workgroup with the shared directory.

NOTE *These commands assume that the name of the Microsoft Windows computer is listed in your DNS server or* `/etc/hosts` *file. You could substitute the IP address for the computer name.*

Samba Terminal Mode

With the name of the Microsoft Windows computer and share, you can connect directly to that shared directory as if it were an FTP server. Once connected, you can upload and download files as well. For example, the command shown in Figure 24.2 connects to the directory I used for this book. Note how I use double quotes with the `cd` command to navigate to a two-word Windows XP directory.

FIGURE 24.2
The direct Samba connection

```
[root@Enterprise3 root]# smbclient //allaccess/RHEL3 -U michael
Password:
Domain=[ALLACCESS] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> cd "Chapter 23"
smb: \Chapter 23\> dir
  .                                   D        0  Tue May  4 11:08:54 2004
  ..                                  D        0  Tue May  4 11:08:54 2004
  .xvpics                             D        0  Tue May  4 11:08:49 2004
  4179c23.doc                         A    72192  Wed Feb 25 12:36:00 2004
  4347c23.doc                         A   173568  Mon May  3 15:00:38 2004
  4347c23.zip                         A   113783  Mon May  3 15:01:24 2004
  example.ldif                        A      162  Sun May  2 12:33:24 2004
  f2301.tif                           A   959522  Wed Apr 28 09:49:40 2004
  f2302.tif                           A   743802  Sun May  2 13:48:10 2004
  f2303.tif                           A   596714  Sun May  2 19:35:04 2004
  slapd.conf                          A     2858  Fri Apr 30 09:45:30 2004
  ~WRL0415.tmp                       AH   135680  Thu Apr 29 23:08:41 2004
  ~WRL1219.tmp                       AH   141824  Fri Apr 30 09:47:00 2004
  ~WRL1310.tmp                       AH   144384  Fri Apr 30 11:57:55 2004

               45778 blocks of size 524288. 12335 blocks available
smb: \Chapter 23\> help
?              altname        archive        blocksize      cancel
cd             chmod          chown          del            dir
du             exit           get            help           history
lcd            link           lowercase      ls             mask
md             mget           mkdir          more           mput
newer          open           print          printmode      prompt
put            pwd            q              queue          quit
rd             recurse        reget          rename         reput
rm             rmdir          setmode        symlink        tar
tarmode        translate      !
smb: \Chapter 23\>
```

Also note the list of available commands. Many of these commands should look similar to the FTP client commands described in Chapter 22. In fact, Samba implements the chroot jail features described in that chapter.

Connecting to a Printer

Shared printers from Microsoft Windows computers should be easy to configure in Linux. If the browse functionality of your Microsoft network is working, you'll be able to select the printer with the Red Hat Printer Configuration tool, in the Queue Type screen, as described in Chapter 20.

But this is not always possible. Microsoft browsing may have trouble finding your printer on a timely basis. Or you may have forgotten to make your printer browsable. You can configure a standard local printer and change the settings later. For example, Figure 24.3 illustrates a standard printer that we configured locally.

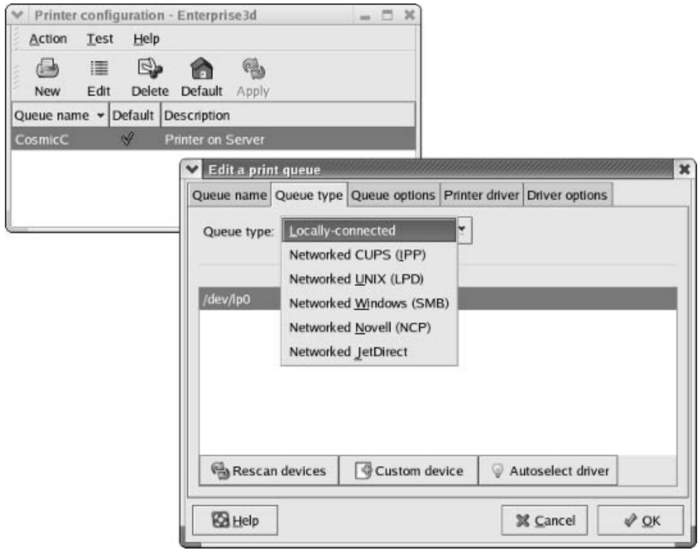


FIGURE 24.3
A local printer in the Printer Configuration tool

It's easy to change this to point to a remote printer; you simply click on the Queue Type drop-down box. From the resulting list, select Networked Windows (SMB) printer.

The format required to connect to a Samba printer is illustrated in Figure 24.4. Each entry is described in Table 24.3.

TABLE 24.3: INFORMATION FOR CONNECTING TO A SHARED SAMBA PRINTER

FIELD	DESCRIPTION
Share	The share name in the `//servername/printername` format.
Server Hostname Or IP Address	The name or IP address of the computer that's sharing the printer.
Workgroup	The Windows workgroup name; enter only if the Windows server is in a workgroup.
User	The Microsoft username of the authorized user.
Password	The Microsoft password associated with the user.

Understanding the Samba Configuration File

With the latest Samba server, there is now one standard Samba configuration file in the `/etc/samba` directory: smb.conf. As we configure Samba, you'll find several more configuration files in this directory related to users, passwords, and security settings. The `/etc/smb.conf` file is a complex file, which we'll examine in detail in this section.

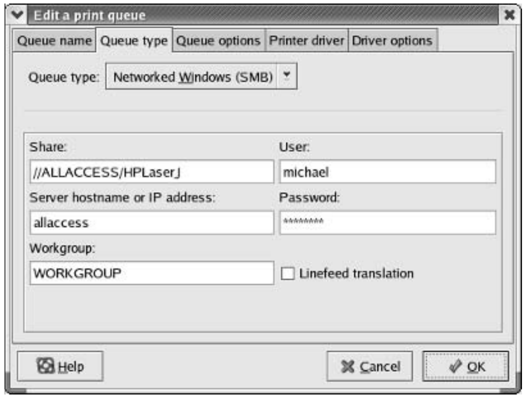


FIGURE 24.4
Configuring a Samba print queue

If you're going to follow along with this book, we suggest that you back up all files from `/etc/samba` in another directory. That way, if you lose track of your changes, you can restore these files without reinstalling the applicable packages.

The `smb.conf` file especially includes a substantial number of useful comments that help you learn to configure Samba. If you're just learning Samba, you should back up `smb.conf` for three reasons:

- The comments in the original `smb.conf` can help you learn more about Samba.
- Tools such as SWAT and `redhat-config-samba` may eliminate some comments when they write changes to `smb.conf`.
- Tools such as SWAT and `redhat-config-samba` may leave out default settings such as `workgroup=WORKGROUP` from your `smb.conf` file.

Samba Daemons

There are two basic Samba daemons: `smbd` and `nmbd`. After changing any configuration file, you should at least reload Samba. When you edit the main Samba configuration file, `/etc/samba/smb.conf`, you need to make Samba read your changes with the `service smb reload` command. However, if you've made any major changes, it's useful to restart both daemons. Restarting the `smbd` daemon with the following command stops and starts both `smbd` and `nmbd` automatically:

```
# service smb restart
```

Other Samba Configuration Files

The other files in the `/etc/samba` directory are `lmhosts`, `secrets.tdb`, `smbpasswd`, and `smbusers`. As we mentioned earlier, they are fairly simple files. Other files can be added during the Samba configuration process.

LMHOSTS

Similar to `/etc/hosts`, the `lmhosts` file is a database of IP addresses and NetBIOS names. A NetBIOS name is a name of a Microsoft Windows computer, typically limited to 15 alphanumeric characters. The default `lmhosts` file includes one line; Microsoft operating systems also use the localhost name to refer to the local computer.

```
127.0.0.1 localhost
```

SECRETS.TDB

The `secrets.tdb` file in this directory normally includes the security identifier (SID) used on a Microsoft Windows network.

SMBPASSWD

The `smbpasswd` file contains the Microsoft Windows network passwords that others can use to log into your local Samba server. The format is similar to the standard Linux authentication file, `/etc/passwd`. There are seven columns of data in this file, separated by colons. We describe each column in Table 24.4.

Table 24.4: Data in */etc/samba/smbpasswd*

Field	Description
Username	Corresponds to an existing username on the Linux computer.
UID	Matches the Linux User ID for the specified user.
LANMAN password hash	LANMAN is an older Microsoft Windows networking password service, associated with Windows 9*x* computers. If you see a series of Xs in this column, the password is disabled.
NT password hash	Associated with Microsoft Windows NT/2000/XP/2003 systems. If you see a series of Xs in this column, the password is disabled.
Account flags	Specifies the type of account; U is user.
Last change time	Specifies the time of the last password change, in seconds, after Jan. 1, 1970.

For detailed information about this file, type the `man 5 smbpasswd` command. (Without the `5`, you'll get the man page for the `smbpasswd` command.).

You can set up your Linux users with Microsoft passwords with the `smbpasswd -a` *`username`* command. For example, if you wanted to add a Microsoft password for user mao, you'd run the following command:

```
# smbpasswd -a mao
New SMB password:
Retype new SMB password:
Added user mao.
#
```

This automatically adds user mao to `/etc/samba/smbpasswd` with the Microsoft password you've just entered.

SMBUSERS

The `smbusers` file is a database of Linux and Microsoft Windows usernames. By default, it includes two lines:

```
root = administrator admin
nobody = guest pcguest smbguest
```

In other words, the Linux root user is mapped to the Microsoft accounts administrator and admin; the Linux nobody user is mapped to the Microsoft accounts guest, pcguest, and smbguest.

It's a straightforward file; if you have a Linux user you want to map to a Microsoft account with a different username, you can add it to this file using the text editor of your choice. For example, if you have a Linux user named elizabeth and a Microsoft user named EZinkann, just add the following line to `/etc/samba/smbusers`:

```
elizabeth = EZinkann
```

NOTE *Samba has discontinued the use of the* `smbadduser` *command with version 3.0.*

This database won't work until you activate the following line in `smb.conf`:

```
; username map = /etc/samba/smbusers
```

In Samba configuration, the hash mark (#) and the semicolon (;) are both used to start comment lines. To activate this line, open `/etc/samba/smb.conf` in a text editor and delete the semicolon from the front of this line. A number of other lines in `smb.conf` include the semicolon; the rest of this chapter explains what happens if you delete various semicolons to activate specific commands.

The Main Samba File: *smb.conf*

The default Samba configuration file, `/etc/samba/smb.conf`, includes a number of comments that make it a rich source of information. However, the comments may be cryptic to those of you who are less familiar with the Samba service. If you haven't already done so, save a copy of this file in another directory.

NOTE *If you've already configured Samba, you may not have the original* `smb.conf` *file with comments. You can get another copy by backing up and then removing your current Samba configuration files from* `/etc/samba` *and then reinstalling the* `samba-common-*` *package with the* `rpm -Uvh --force samba-common-*` *command. Don't forget to restore your original Samba configuration files when you're done.*

The `smb.conf` file includes *global* settings for connecting to a desired Microsoft Windows–based network. It also includes *share* definitions for any directories and printers that you may want to share with other computers on your LAN. Different groups of settings help you work in a LAN that's configured as a Microsoft *workgroup*, as a member server, or even as a primary or backup domain controller.

The following sections include a basic analysis of the standard settings in `/etc/samba/smb.conf`, in order. Later in this chapter, you'll use SWAT and the Samba Server Configuration tool (`redhat-config-samba`) to configure `smb.conf`.

The following section analyzes the `smb.conf` file from the Red Hat `samba-*` RPM package. Many of the settings in this version of `smb.conf` vary from the Samba defaults.

NOTE *I've included tips if you want to configure this computer as a Microsoft Windows NT 4–style PDC. In fact, Samba allows you to substitute a Linux computer for that type of server, which can save you the costs of upgrading to Microsoft Windows 2003 server.*

Samba Global Settings

The `smb.conf` file contains a substantial number of `[global]` variables. If you don't use a variable, Samba will assume the default for that variable.

With different global variables, you can:

- Limit the IP addresses allowed to access your server.
- Set printers as a part of the Samba browse list.
- Configure guest accounts and log files.
- Configure Samba to match the predominant Windows security mode on your network.
- Take advantage of the many password settings available.
- Map Linux usernames to Windows usernames.
- Customize configuration files for different computers.
- Limit Samba authentication by using Pluggable Authentication Modules (PAM), as described in Chapter 17.
- Configure Samba to send data in different-sized data chunks and through different interfaces.
- Set the browse list, where shared information is advertised, possibly based on different master computers on a network.
- Make Samba conform to the logon parameters on a Microsoft network.
- Store profiles on a Linux computer.
- Set up Samba to work with WINS and DNS.

Linux is case sensitive, and Windows is not; Samba helps you bridge the difference.

NOTE *WINS is short for the Windows Internet Name Service, which keeps a database of Microsoft-style NetBIOS names and IP addresses. While it functions like a DNS server, its scope is generally limited to a LAN.*

Basic Network Type

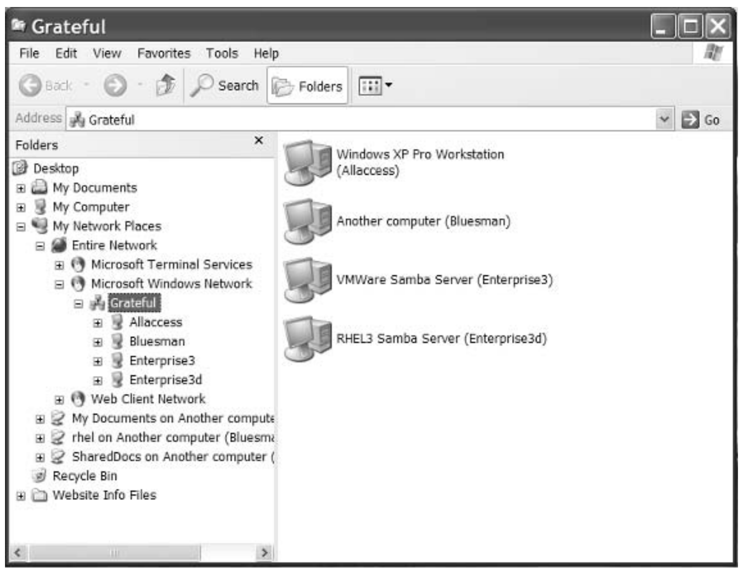
The first global variable describes the type of network that you're trying to join. While the name of the variable is `workgroup`, you can set it to the name of your Microsoft network's workgroup or domain. For example, if your network's domain is named grateful, substitute the following line in `smb.conf`:

```
workgroup=grateful
```

Samba also can include a description of your computer; Figure 24.5 reflects the following command for the computer named Enterprise3d:

```
server string = RHEL3 Samba Server
```

FIGURE 24.5
A Microsoft Windows view of a shared Samba server



TIP *One command in* `smb.conf` *can help the Microsoft network system find your Linux computer. This is especially helpful if you're setting up a Windows NT 4–style domain. For example, if your computer is named enterprise3, the corresponding command is* `netbios name = enterprise3`.

IP Address Limits

You can limit access to Samba through the appropriate firewall commands described in Chapter 17. You can further limit access with the `hosts allow` command. For example, either of the following commands limits access to the local computer and the 10.122.33.0 network:

```
hosts allow = 10.122.33. 127.
hosts allow = 10.122.33.0/255.255.255.0 127.
```

Samba and Printers

By default, printers are included in the list of shared, browsable items. The following commands load the list of printers from `/etc/printcap` for a standard CUPS-based system:

```
printcap name = /etc/printcap
load printers = yes
printing = cups
```

NOTE *There are a number of parameters in Samba that look like they are misspelled. They may still be good. For example,* `browsable` *works as well as* `browseable`*, and* `writable` *works as well as* `writeable`*.*

Guest Accounts

Samba lets you create a standard guest account. For example, if you're setting up a workstation for people in a lobby, you may want them to access your advertising but nothing else. If you activate the following standard command, make sure that pcguest is a real user on your Linux system:

```
; guest account = pcguest
```

TIP *Remember, a semicolon in front of a command "comments it out"; you'll need to delete the semicolon to activate the command.*

The pcguest option is important, especially if you're configuring a Microsoft domain, where directories can be shared only with real users.

Log Files

The following option configures different log files for each computer that connects to your Samba server. For example, if you have a Windows computer named Havel, the following line means you can find debugging information in `havel.log` in the noted directory. A `max log size` of 0 means that there is no limit on the size of these log files; other limits are in kilobytes.

```
log file = /var/log/samba/%m.log
max log size = 0
```

NOTE *Any expression in* `smb.conf` *that starts with a* `%` *can vary. For example,* `%m` *represents the name of the client computer and thus changes depending on the client.*

Security Modes

There are several basic security modes on Microsoft Windows networks. Generally, what you select is based on the conditions for the shared directory and the type of shared network. The options are described in Table 24.4.

```
security = share
security = user
security = server
security = domain
security = ads
```

Table 24.5: Samba Security Modes

Mode	Description
share	For systems where shared directories do not require anything more than a password for access; most common for workgroups of peer-to-peer computers without any dedicated servers.
user	For systems where shared directories are limited by usernames and passwords; common to server-level computers such as Windows 2000, Windows XP, and, yes, Linux, on a peer-to-peer workgroup network. Believe it or not, if you're setting up a Windows NT 4 PDC, you'll want to set up `security = user`.
server	For systems where usernames and passwords prefer a centralized database not associated with a domain; if such a database cannot be found, this reverts to `security = user`.
domain	For systems that are connecting to a Windows-style domain; requires `smbuser` and `smbpasswd` database files in `/etc/samba`.
ads	For systems that are connecting as a Member Server in a Windows 2000/2003 Active Directory–style domain.

Password Settings

Several password settings are available in Samba. If you're configuring a central server for Microsoft Windows usernames and passwords, you can specify it here. The PDC can even be located on a Samba-enabled Linux computer.

If you have set `security = server` or `security = domain`, you should also specify the password servers for the network. For example, if you know that the names of your PDC and BDC are ntserv1 and ntserv2, you could insert the following command:

```
password server = ntserv1 ntserv2
```

Or, if you don't know the names of your PDC or BDC, the following command sets your Samba server on a search for domain controllers:

```
password server = *
```

Several older Microsoft Windows operating systems don't work very well on passwords with mixed upper- and lowercase characters. The commands, if active, try all combinations of upper- and lowercase characters on an eight-character password and username.

```
; password level = 8
; username level = 8
```

Normally, Samba is configured to send encrypted passwords from the standard Samba passwords file. Remember, this password file includes Microsoft Windows usernames and passwords that you added with the `smbpasswd -a` command. However, not all Microsoft Windows computers can handle encrypted passwords.

```
; encrypt passwords = yes
; smb passwd file = /etc/samba/smbpasswd
```

Without these commands, Samba would revert to the default, sending passwords over the network in clear text. That's still required for the first versions of Microsoft Windows 95 and earlier Microsoft operating systems.

If users change their passwords on a Microsoft Windows computer, the following commands synchronize the corresponding Linux password:

```
unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n *passwd:
➥ *all*authentication*tokens*updated*successfully*
```

Mapping Linux and Windows Users

As described earlier, you can match your Linux and Windows users with different usernames, storing the corresponding names in `/etc/samba/smbusers`. If you plan to use this database, activate the following command:

```
; username map = /etc/samba/smbusers
```

Customizing Samba by Computer

You can configure Samba servers on remote computers. If you activate the following command, each computer will look for a specific configuration file. For example, if your Windows computer name is Jamco, the `%m` variable makes it look for the `/etc/samba/smb.conf.Jamco` configuration file when it connects.

```
; include = /etc/samba/smb.conf.%m
```

Performance Management

When you're more comfortable with Samba, you'll learn to optimize network performance. What you do depends on the size and traffic on your network. In the following command, `TCP_NODELAY` often doubles Samba performance. The `SO_RCVBUF` and `SO_SNDBUF` variables are buffers for data coming in and out of Samba. Optimal settings vary with the load on your Samba server. If you want to experiment, adjust each by 1KB (for example, `SO_RCVBUF=7168` or `SO_RCVBUF=9216`).

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

Network Interfaces

Servers can be configured with multiple network interface cards. You can limit Samba access to one network card, or you can set a Samba server to work with a specific remote network. For example, the following line sets Samba to work with the `eth1` network interface card and the 172.168.33.0 IP network address:

```
interfaces eth1 172.168.33.0/24
```

Browsing

On a Microsoft Windows–based network, *browsing* is the ability of computers to see available shared directories and printers. One computer is selected as a browse master; other computers with shares send their information to that computer.

You can even set your Samba server to send its shares to a remote network. If you don't know the IP address of the master browser on that network, just use the broadcast address. For example, the following command synchronizes browse lists between your LAN and the 192.168.1.0 network:

```
remote browse sync =  192.168.1.255
```

This command just sends your Samba server's browse information to that network (alternatively, you can specify the IP address of the browse master computer):

```
remote announce = 192.168.1.255
```

One computer on a Microsoft network keeps the browse list. An "election" is held to determine that computer; even a Samba server can be elected to maintain the browse list. However, the following command keeps the Samba server out of the election:

```
; local master = no
```

If you want your Samba server to participate in a browse election, you can fix its chances with the following command. At this level, Samba will normally win a browse election against any computer but a domain controller or a Microsoft Windows NT server:

```
; os level = 33
```

If you don't want to leave anything to chance, you can set the `os level` to 64 or higher and set Samba to be the master browser for your domain.

```
; domain master = yes
```

If your Samba server is underworked, you may want to set it to be the preferred master browser with the following command:

```
; preferred master = yes
```

Logon Management

If you have Linux and Microsoft Windows computers on your network, you can set Samba to control the username and password database as a PDC for that network by activating the following command:

```
; domain logons = yes
```

NOTE *This also requires user-level security and a* `[netlogon]` *directory, which are described in other parts of the Samba* `smb.conf` *file.*

A Microsoft network lets you configure logons by user or by computer. Each is configured by a logon script, which you can store on your Samba server. `%m` corresponds to each computer (machine), and `%U` corresponds to each user.

```
;  logon script = %m.bat
;  logon script = %U.bat
```

With a centralized profile, logons by user can provide a consistent look and feel for that user on any Microsoft computer on that network. You can store the profiles on your Samba server, in the logon path. `%L` represents the name of the server; while `%U` is the username.

```
; logon path = \\%L\Profiles\%U
```

If you activate this command, the result varies depending on the type of Microsoft Windows client. If the client is a Windows 9*x*/ME computer, profiles are stored in each user's home directory. If the client is a Windows NT/2000/XP computer, profiles are stored in the `/home/profiles/$USER` directory.

WINS and DNS

The Windows Internet Name Service (WINS) is similar to DNS, except that it is a database of NetBIOS names and IP addresses. If Samba isn't able to find the name of a computer in `/etc/hosts`, WINS and DNS provide two alternative databases.

TIP *If you're setting up a Windows NT 4 PDC, it helps to set up WINS on the PDC computer.*

The following command sets up WINS on the local Samba server:

```
; wins support = yes
```

Alternatively, you can look to a WINS server on a specific IP address (the address shown is arbitrary; substitute appropriately). In this case, your Samba server becomes a WINS client:

```
; wins server = 192.168.0.22
```

If your Microsoft-based network includes older computers, you may want to activate this command to allow all computers access to the WINS database:

```
; wins proxy = yes
```

Or, if the computer is not in the WINS database, you can set up your DNS server as an alternate database by activating this command:

```
; dns proxy = yes
```

Domain Users and Groups

If you're configuring a Samba server as a Windows NT 4–style PDC, you'll want to assign a range of user and group ID numbers that doesn't interfere with any Linux or Unix UIDs and GIDs on the

local network. Remember, standard Linux UID and GID numbers start with 500. The following commands assign a fairly high range for domain user and group ID numbers:

```
idmap uid = 5000-10000
idmap gid = 5000-10000
```

Older versions of Samba used the `winbind uid` and `winbind gid` commands for this purpose.

TIP *If you're configuring the local computer as a Windows NT 4–style PDC, you'll need to activate the* `winbindd` *daemon, which you can activate with the* `service winbind start` *command.*

Case Management

Linux is a case-sensitive operating system; Microsoft operating systems are not. Normally, Samba preserves the case of transferred files. You can force everything into lowercase; the following commands affect long filenames and filenames that follow the old Microsoft 8.3 filename format (for example, abcdefgh.123):

```
; preserve case = no
; short preserve case = no
```

In contrast, you can set all files to default to lowercase with the following command:

```
; default case = lower
```

If all your users are disciplined about case-sensitive filenames on all computers on your network, you may be able to make your Samba server case-sensitive too with this command:

```
; case sensitive = yes
```

Remember, Microsoft Windows is not a case-sensitive operating system; if you activate case-sensitivity, any mistakes in the case of various filenames can cause problems.

NOTE *Configuring Samba as a PDC is a rich and complex topic, which itself could fill a book this size. For more information, review the latest Samba-3 HOWTO and Reference Guide, available online at* `us1.samba.org/samba/docs/man/howto`.

DEFAULT GLOBAL SETTINGS

Default settings for global variables are listed in Table 24.5. Remember, if you use a default parameter, you don't even need to include it in `smb.conf`; tools such as SWAT and the Samba Server Configuration tool (`redhat-config-samba`) will delete it when you use them to update `smb.conf`.

TABLE 24.6: DEFAULT *SMB.CONF* GLOBAL SETTINGS

VARIABLE	DEFAULT
`case sensitive`	no
`default case`	lower

Continued on next page

Table 24.6: Default *smb.conf* Global Settings *(continued)*

Variable	Default
dns proxy	yes
domain logons	no
encrypt passwords	yes
guest account	nobody
hosts allow	none (all hosts allowed access)
inherit permissions	no
interfaces	All active interfaces except 127.0.0.1 (if you can send a broadcast message to that address)
load printers	yes
local master	yes
logon path	\\%N\%U\profile, where %N is the NIS server and %U is the username
max log size	5000 (KB)
min password length	5
name resolve order	lmhosts host wins bcast
obey pam restrictions	no
pam password change	no
passwd chat	*new*password* %n\n *new*password* %n\n* changed
password level	0
preferred master	auto
preserve case	yes
printcap name	/etc/printcap
security	user
server string	Samba %v, where %v is the version number
short preserve case	yes
socket options	TCP_NODELAY
unix password sync	no
username level	0
wins proxy	no
wins server	Not enabled
wins support	no
workgroup	WORKGROUP

Configuring a Share

Now it's time to analyze the way directories are shared from the packaged `smb.conf` configuration file. There are seven examples of shared directories in the standard `smb.conf` file; once we examine each of these examples, you'll have a much better idea of how to configure your own shared directories.

The stanzas I describe here aren't in the same order as you'll find in the default `smb.conf` file. Instead, I've grouped similar types of shared directories together.

The [homes] Share

Microsoft Windows users with accounts on your Linux computer can get read and write access to their own home directories. All you need is the following standard commands in `smb.conf`:

```
[homes]
   comment = Home Directories
   browseable = no
   writeable = yes
```

Implicit in a `[homes]` share are the following defaults. While there is normally no default list of `valid users`, home directories are normally visible only if the Microsoft user has the same account info on the Samba server. The other two variables are standard defaults.

```
   create mode = 0744
   directory mode = 0755
```

These commands are explained in Table 24.6.

Table 24.7: Typical Samba Home Directory Share Commands

Command	Description
`[homes]`	This is a standard "special" section in `smb.conf`.
`comment = Home Directories`	This command describes the share for Windows Network Neighborhood, My Network Places, or `smbclient -L \\hostname`.
`browseable = no`	Normally, `browseable=no` keeps the shared directory from being shown in Network Neighborhood or My Network Places; this does not apply for users' own home directories.
`writeable = yes`	This command allows users to write to that directory; you can also use `read only=no`.
`create mode = 0744`	This command sets `rw-r--r--` permissions on new files. It does not override permissions set on Windows NT/2000/XP computers. It's also known as `create mask`.
`directory mode = 0755`	This command sets `rwxr-xr-x` permissions on new directories. It does not override permissions set on Windows NT/2000/XP computers. It's also known as `directory mask`.

To get to their directory from a Microsoft Windows computer, users simply must enter their Linux username and password in the Connect To *Computername* window, shown in Figure 24.6.

FIGURE 24.6
User michael connects to a shared Samba home directory

The [tmp] share

You can set up the `/tmp` directory as a common place for users on your network to share files. The following commands set it up as accessible for any user:

```
[tmp]
   comment = Temporary file space
   path = /tmp
   read only = no
   public = yes
```

These commands are straightforward; the comment is added to the Windows Network Neighborhood or My Network Places view of `/tmp`; any valid user can write to this directory. The `public = yes` command is new and is synonymous with `guest ok = yes`. In other words, a password is not even required.

The [public] Share

You don't need to share directories with everyone. Similar to the User Private Group scheme described in Chapter 9, you can set up a directory that's readable to all but writeable only by users in the group named staff.

```
[public]
   comment = Public Stuff
   path = /home/samba
   public = yes
   writable = yes
   printable = no
   write list = @staff
```

Before you set up this particular share, you need to make sure there is a `/home/samba` directory, as well as a staff group, in `/etc/groups` and `/etc/gshadow`.

Another [public] Share

One variation may be useful for more public situations; the commands that follow configure a directory where all files are readable and writeable by all users. However, the `only guest = yes` command means that any user who connects to this directory has only the privileges of the guest user. Of course, you need to make sure that the `path` directory—in this case, `/usr/somewhere/else/public`—actually exists.

```
[public]
   path = /usr/somewhere/else/public
   public = yes
   only guest = yes
   writable = yes
   printable = no
```

A Share for Two

One more variation configures a share with just two valid users—in this case, Mary and Fred. While it isn't a public share, you'll see later that `browseable = yes` by default. In other words, other users can see Mary and Fred's share, but they can't access their shared directory unless they have one of their usernames and passwords.

```
[myshare]
   comment = Mary's and Fred's stuff
   path = /usr/somewhere/shared
   valid users = mary fred
   public = no
   writable = yes
   printable = no
   create mask = 0765
```

Remember, the items noted have to exist on the Samba server. In this case, that includes the `/usr/somewhere/shared` directory and the users named mary and fred.

A Private Directory

You can configure a private directory other than their home directory for individual users. For example, the following commands sets up a private directory, `/usr/somewhere/private`, for the Linux user named fred. Since `public = no`, guest users are not allowed to access this directory.

```
[fredsdir]
   comment = Fred's Service
   path = /usr/somewhere/private
   valid users = fred
   public = no
   writable = yes
   printable = no
```

A Shared Directory for a Computer

You can configure a directory just for a specific computer. This can be quite useful for different users on the same computer. For example, it's a good place for someone in a factory to leave information for his or her counterpart on a different shift.

```
[pchome]
   comment = PC Directories
   path = /usr/local/pc/%m
   public = no
   writable = yes
```

You just need to create the directory listed as the `path`. Remember, `%m` represents the name of the computer. For example, if a computer named factory1 is trying to connect, the previous `path` command means that you need to create a `/usr/local/pc/factory1` directory.

Sharing a Printer

If you've configured CUPS printers, you still need to configure the basic share. Even though the standard `smb.conf` file suggests the BSD-style print system, the following commands work with CUPS printers as well:

```
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   writable = no
   printable = yes
```

If you have a single LPD printer you want to share, a different preconfigured share is available in the standard `smb.conf` file to provide exclusive use—in this case, to the user named fred:

```
[fredsprn]
   comment = Fred's Printer
   valid users = fred
   path = /home/fred
   printer = freds_printer
   public = no
   writable = no
   printable = yes
```

The limit implied by `writable = no` does not affect print spool directories; your computer can still send print spool files to the print server.

Configuring Logon Directories

When you use Samba to configure your Linux computer as a domain controller on a Microsoft network, you need to configure logon and profile paths for each user. As before, the directories shown must already exist.

The following commands can configure logons to a Microsoft Windows–style domain, based on the directory specified by `path`:

```
[netlogon]
   comment = Network Logon Service
   path = /usr/local/samba/lib/netlogon
   guest ok = yes
   writable = no
   share modes = no
```

This is one directory where you may want `writable = no` and `share modes = no`; otherwise, users may rewrite their own logon scripts, and a cracker may figure out how to get every user's logon information. Speaking of denying information to crackers, you may want to keep them away from the names of the netlogon files, which include usernames. You can do this by adding one more command to this stanza:

```
browseable = no
```

TIP *Administrators who configure Samba as a Microsoft Windows PDC may want to use an easier* `path`*, such as* `/home/netlogon`*.*

The following commands can configure profiles locally for users who log into your Samba server as if it were a Microsoft Windows server:

```
[Profiles]
   path = /usr/local/samba/profiles
   browseable = no
   guest ok = yes
```

Microsoft Windows networks often allow users to configure their own roaming profiles. Assuming you've activated the `logon path` command in the `[global]` section, you can set a logical path for Windows NT/2000/XP profiles, such as:

```
path = /home/profiles
```

Naturally, this directory should exist. To allow users to change their own profiles, you'll need to configure it with 777 permissions. While this may sound odd, access is still limited to profile owners based on the following commands:

```
create mode = 0600
directory mode = 0700
```

And to promote security, the following commands keeps crackers and others who may be curious from looking through user profiles:

```
browsable = no
guest ok = no
```

Default Share Settings

Default settings for shared directories and printers are listed in Table 24.7. Remember, if you use a default parameter, you don't even need to include it in `smb.conf`; tools such as SWAT and `redhat-config-samba` delete default settings in `smb.conf`.

Table 24.8: Default *smb.conf* Share Settings

Variable	Default
`browseable`	yes
`comment`	No default
`create mode`	a.k.a. `create mask = 0744`
`directory mode`	a.k.a. `directory mask = 0755`
`guest ok`	no
`path`	No default
`printable`	no
`public`	a.k.a. `guest ok = no`
`read only`	yes
`writable`	no (the true default is `read only = yes`)
`write list`	No default (any standard user can write to a specified share)
`valid users`	No default (any standard user can connect to a share)

A Samba Troubleshooting Checklist

Samba configuration files, especially `smb.conf`, can be quite large. Small errors can throw a monkey wrench into your service. It's easy to spend a few hours revising your configuration when the problem is as simple as an extra firewall.

When troubleshooting, the first thing you should do is check the syntax of the `smb.conf` file. Pay particular attention to comments; it's common to accidentally delete a comment code such as `;` or `#`. Next, you should check the browse list from the local Samba server. If the local browse list is good, take a careful look at your network. And a number of valid `smb.conf` settings can cause problems.

Testing *smb.conf*

Once you've configured `smb.conf`, it's easy to test. The `testparm` command acts as a syntax checker for your Samba configuration file. If you don't specify the location, `testparm` automatically checks the `smb.conf` file in the `/etc/samba` directory.

Before restarting or reloading the `smb` daemon, run `testparm`. If you've made a small mistake in editing, it can point you right to the source of the problem `smb.conf`, which can save you a lot of grief.In Figure 24.7, I've illustrated this process on a Samba server that I've configured as a PDC.

FIGURE 24.7
Testing a Samba configuration file

```
[root@Enterprise3d root]# testparm | more
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[dos]"
Loaded services file OK.
Server role: ROLE_DOMAIN_PDC
Press enter to see a dump of your service definitions
# Global parameters
[global]
        workgroup = GRATEFUL
        server string = RHEL3 Samba Server
        username map = /etc/samba/smbusers
        log file = /var/log/samba/%m.log
        max log size = 50
        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
        domain logons = Yes
        os level = 64
        preferred master = Yes
        domain master = Yes
        dns proxy = No
        wins support = Yes
        idmap uid = 5000-10000
        idmap gid = 5000-10000

[homes]
        comment = Home Directories
        read only = No
        browseable = No

--More--
```

Checking the Local Samba Browse List

Once you've restarted Samba, check the list of what you're sharing by using the `smbclient` command. If you see the right list on the Samba server, you should be able to see the same list on Microsoft Windows computers on your network, barring a network- or firewall-related problem. For example, the following command checks the list on the computer named Enterprise3d; michael is a user on that computer:

```
# smbclient -L \\Enterprise3d -U michael
```

You're prompted for mj's password, and then you should see the browse list for the Enterprise3d computer. In the example shown in Figure 24.8, you can also see the members of the Domain named GRATEFUL and a connection the workgroup named WORKGROUP.

Checking Your Network

As discussed in Chapter 16, most network problems are physical; you may have a problem with a loose cable, no power on a hub, or a similar issue. We examined a number of commands in Chapter 16, such as `ping` and `netstat`, that can help you check the status of a network.

One problem I often run into is firewalls. If there's a firewall on the Samba server, it can block communication with clients. If your Samba server can't see clients, you won't be able to log onto a shared Samba directory or printer.

Figure 24.8
Checking a local Samba server

```
[root@Enterprise3d root]# smbclient -L \\Enterprise3d -U michael
Password:
Domain=[GRATEFUL] OS=[Unix] Server=[Samba 3.0.2-6.3E]

        Sharename       Type      Comment
        ---------       ----      -------
        dos             Disk      DOS drive
        IPC$            IPC       IPC Service (RHEL3 Samba Server)
        ADMIN$          IPC       IPC Service (RHEL3 Samba Server)
        CosmicC         Printer
        michael         Disk      Home Directories
Domain=[GRATEFUL] OS=[Unix] Server=[Samba 3.0.2-6.3E]

        Server               Comment
        ---------            -------
        ALLACCESS            Windows XP Pro Workstation
        BLUESMAN             Another computer
        ENTERPRISE3          VMWare Samba Server
        ENTERPRISE3D         RHEL3 Samba Server

        Workgroup            Master
        ---------            -------
        GRATEFUL
        WORKGROUP            ENTERPRISE3M
[root@Enterprise3d root]#
```

Other Samba Issues

I've encountered other problems with Samba, mostly related to mistakes that I've made in the `smb.conf` configuration file. Some mistakes are valid options, as they'll pass a `testparm` syntax check, but they won't represent your Samba server properly on your LAN. Sometimes you can get clues from the applicable log file. As described earlier, there are default log files specific to each Samba client. For example, Figure 24.9 lists connections from the computer named enterprise3m.

Common mistakes you can make in `smb.conf` fall into the following areas:

The wrong workgroup The default `workgroup` on Microsoft Windows 2000/XP computers is MYGROUP. This differs from the default value of `workgroup`, which is WORKGROUP. The problems get worse if you're setting up this computer on a domain and don't enter the right domain name for this variable.

browsable = no If you set `browsable = no`, users normally will not be able to see your shared directory or printer in their Windows Network Neighborhood or My Network Places.

Improperly configured sharing As you've seen in previous sections, there are a number of ways to share—with users, guests, groups, or everyone. If sharing is not properly configured, your users may not be able to get to the directories or printers they need.

writable = no Samba shared directories are read-only by default. If you don't specify otherwise, your users won't be able to write to appropriate shared directories.

Improperly configured firewalls Standard Red Hat Enterprise Linux firewalls block Samba communication. If you have a `hosts allow` variable, computers not on the list can't get to your Samba server.

Figure 24.9
Samba log file with client connections

Managing Samba Users and Computers

We can set up a different database of users for Microsoft Windows networking through Samba. Samba usernames and passwords do not have to be identical to the usernames and passwords you may use to log in directly to a Linux computer. The relationship is built into the `smbusers` and `smbpasswd` configuration files in the `/etc/samba` directory.

On a Microsoft network, any computer that is a member of a domain requires its own account. We'll show you how to set this up in `/etc/passwd` and `/etc/samba/smbpasswd`. Once complete, you can then configure Microsoft and Linux computers to join the Samba-based domain.

This section in part summarizes the commands associated with Samba. We covered several of these commands earlier in this chapter. The following sections are focused on setting up a single database of usernames and passwords on a Microsoft Windows NT 4–style network, which makes it possible for you to set up a Red Hat Enterprise Linux 3 sever as a substitute (and upgrade) to that older server.

Configuring Computer Accounts

Those of you who are more familiar with Microsoft networking on domains probably already know that Microsoft computer accounts include a $ at the end. This also works when you configure a computer account on a Linux computer acting as a PDC on a Microsoft network. For example, the following line from my Linux PDC's `/etc/passwd` file is the computer account for the allaccess computer on my domain:

```
allaccess$:x:503:100::/dev/null:/bin/false
```

This configures allaccess with a User ID of 503, a common group ID of 100, with no home directory (`/dev/null` is the standard Linux trash bin), and no login shell (`/bin/false` keeps crackers from using this account to actually log into a Linux computer).

You can also set up this entry with the following command:

```
# useradd -d /dev/null -g 100 -s /bin/false -M allaccess
```

Once you set up a basic account, you can join a computer to a Microsoft domain. As this is not a book on Microsoft operating systems, I won't go into details. However, you'll be prompted for an administrative account on the PDC from the client. For example, when I join a domain from a Windows XP Professional computer, I'm prompted for this information as shown in Figure 24.10.

FIGURE 24.10
Joining a client to a domain

If you're joining a Microsoft domain from a Linux computer, you'll want to get the Microsoft SID identifier for the domain. I've done so on my enterprise3 computer with the following command:

```
# net rpc getsid
Storing SID S-1-5-21-3316416275-723232865-759781495 for Domain GRATEFUL
  ➥in secrets.tdb
```

Then you can join the domain easily enough. Since the administrative user on the domain is also root, all you'll need to do is enter the root password on the PDC.

```
# net rpc join
Password:
Joined domain GRATEFUL
```

Samba Management Commands

While Red Hat Enterprise Linux 3 includes good GUI tools for managing Samba servers and clients, many administrators believe that it is better to manage Samba from the command-line interface. As with other GUI tools, they are "front ends" to what you can run at the command line.

However, we believe that unlike others, the Samba-related GUI tools are of the highest quality and can help harried administrators administer a Samba server more efficiently. As space is limited, we can only summarize the Samba commands in this section. For more information, it's helpful to start with the man page for each command.

nmbd The NetBIOS name server supports the Microsoft standard computer naming service associated with NetBIOS on TCP/IP networks.

The Samba *net* Commands

One of the advantages of Samba 3.0.*x* is the `net` commands. While not identical to the `net` commands from a MS-DOS command-line interface, in our opinion, they actually provide finer-grained control. There are over 50 different variations on the `net` command. This is just a very brief overview of these commands.

```
[root@Enterprise3 root]# net

Usage:

Usage:
  net time              to view or set time information
  net lookup            to lookup host name or ip address
  net user              to manage users
  net group             to manage groups
  net groupmap          to manage group mappings
  net join              to join a domain
  net cache             to operate on cache tdb file
  net getlocalsid [NAME]        to get the SID for local name
  net setlocalsid SID   to set the local domain SID
  net changesecretpw    to change the machine password in the local secrets data
base only
                        this requires the -f flag as a safety barrier
  net status            Show server status

  net ads <command>     to run ADS commands
  net rap <command>     to run RAP (pre-RPC) commands
  net rpc <command>     to run RPC commands

Type "net help <option>" to get more information on that option
Valid targets: choose one (none defaults to localhost)
        -S or --server=<server>         server name
        -I or --ipaddress=<ipaddr>      address of target server
        -w or --workgroup=<wg>          target workgroup or domain
```

You can get a good summary of the options when you type the `net` command, as shown in the previous graphic. Most of the options shown include several options, such as `net status sessions` for a list of open shared connections and `net status shares` for a list of shared directories.

While there's no direct analogue to the MS-DOS `net view` command, the `net rap server` command does list Microsoft and Samba servers on a CIFS network. The `net rpc` commands are most useful; variations allow you to manage users and groups, list open files and connected computers, join a domain, or even shut down a remote Samba server.

smbcacls Samba supports Microsoft NT–style Access Control Lists. The `smbcacls` command allows you to specify ownership, rights, and permissions on individual files.

smbclient The `smbclient` command supports FTP-style access to Samba servers as well as a detailed view of shared directories and printers from specified servers. We've explained this process in some detail earlier in this chapter.

smbcontrol The `smbcontrol` command lets you send short messages to Samba servers, with respect to synchronizing databases, forcing browser elections and more.

smbcquotas With the `smbcquotas` command, Samba allows you to manage the quotas associated with shared directories from Windows 2000/XP/2003 computers formatted to NTFS 5 standards.

smbd The server associated with the Samba service is `smbd`. It's normally started through the `smb` script in the `/etc/rc.d/init.d` directory.

smbmnt The `smbmnt` command helps the `smbmount` command actually mount directories shared on a Microsoft style—network. While you may never use this command directly, you'll need to set the SUID bit to allow regular users to mount shared Microsoft-style directories with the `smbmount` command.

smbmount If you want to `mount` a Microsoft or a Samba-shared directory, you can do so with the `smbmount` command. This serves as a front end and is equivalent to the `mount.smbfs` command.

smbpasswd Samba 3.0.*x* has expanded the use of the `smbpasswd` command. As we've explained earlier, the `smbpasswd -a` *`username`* command allows you to set up passwords for use on a Microsoft Windows network.

smbspool You can send print jobs directly to a printer shared through a Microsoft Windows network with the `smbprint` command.

smbstatus As an administrator, you may want to monitor the users who are connecting to your Samba servers; this is possible with the `smbstatus` command.

smbtar If you want to back up a shared Microsoft directory onto a Linux system, you can use the `smbtar` command. Unfortunately, the switches associated with this command differ significantly from the `tar` command with which Linux administrators are familiar.

smbtree If you want to review the computers and the shared directories and printers from the command line, use the `smbtree` command.

winbindd To set up a database of computer names on a Microsoft network, you may need a WINS server. If you want to set it up on a Samba computer, you'll need to activate the `wins support = yes` command in `smb.conf` and start the `winbindd` daemon. This daemon is normally started with the `winbind` script in the `/etc/rc.d/init.d` directory.

Using the Samba Web Administration Tool (SWAT)

The generic all-in-one GUI configuration utility associated with Samba is known as the Samba Web Administration Tool (SWAT). As we mentioned earlier, it's included in the `samba-swat-*` RPM package. While it is not included with the Red Hat Enterprise Linux 3 CDs, you can download the binary RPM for this package with a valid subscription to the Red Hat Network or from one of the "rebuild" download sites. Before you can run SWAT, you must activate the corresponding `xinetd` daemon with the following command:

```
# chkconfig swat on
```

SWAT includes several menus, which we'll look at in the following sections. Briefly, once you make your desired configuration changes in each menu, you'll click the Commit Changes button to write the changes to file.

Once you've completed the changes, you must restart the `smbd` and `nmbd` daemons, either through the SWAT Service menu or with the `service smb restart` command.

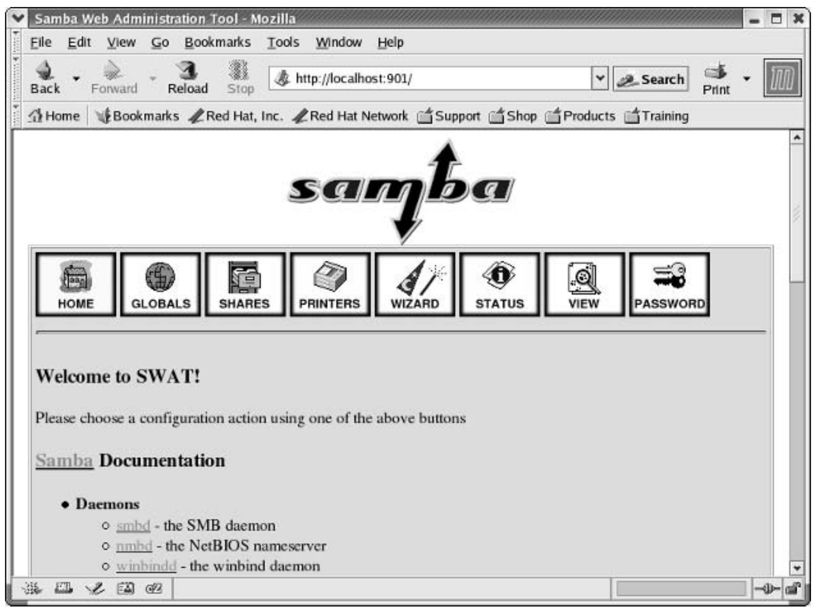
TIP *Before you use SWAT for the first time, back up the files in the* `/etc/samba` *directory, especially* `smb.conf`*. SWAT overwrites the comments in the default version of this configuration file; these comments can help the Linux administrator who is less familiar with Samba.*

One of the weaknesses of SWAT, as included with Red Hat Enterprise Linux, is the lack of data with respect to individual printers. Red Hat has moved printer database packages to its Samba Server Configuration utility. If you want the latest version of SWAT with the full foomatic printer database, as described at `www.linuxprinting.org`.

The Home Menu

To start SWAT, open the browser of your choice and navigate to `localhost:901`. Even if you're logged in as the root user, SWAT prompts you for an authorized username and password. Once you enter that information, SWAT starts with the Home menu, shown in Figure 24.11.

FIGURE 24.11
The SWAT Home menu

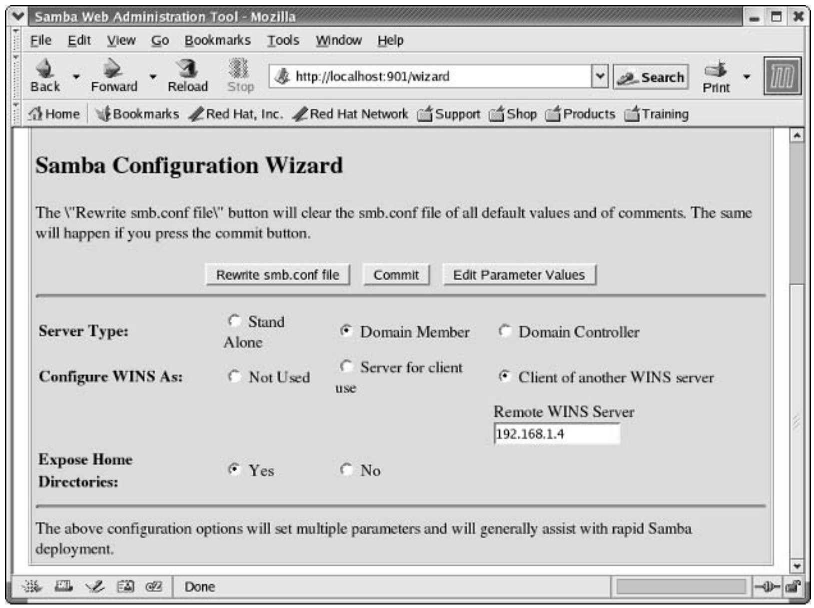


As you can see, the SWAT Home menu includes links to other SWAT menus on the top row. We'll examine each menu in the following sections. The Home menu also includes access to a number of Samba documents. Some are man pages associated with specific commands or files. Toward the bottom of the document list are Samba HOWTOs and a copy of the second edition of *Using Samba*, in e-book format.

Samba Configuration Wizard

SWAT's Samba Configuration Wizard provides a way to address three basic settings for your Samba server. Click the Wizard link, and scroll down to see the options shown in Figure 24.12. Let's take a look at each of these options.

FIGURE 24.12
The Samba Configuration Wizard



Server Type Allows you to select from one of three types of servers on a Microsoft-based network: Stand Alone, Domain Member, or Domain Controller (PDC). If you're connecting to a Microsoft peer-to-peer workgroup, stand-alone servers are your only valid option.

Configure WINS As Lets you specify the role of WINS on your network. There are three choices: You can configure your Samba server as a WINS server; you can configure it as a client of a different WINS server, if you know its IP address; or you can avoid the use of WINS on your network. This is associated with the `wins server` and `wins support` variables.

Expose Home Directories Permits users to see the directories associated with their Linux usernames. This option is associated with the `[homes]` share we described earlier in this chapter.

Once you've made your selections, click the Commit button. This may not be the configuration wizard that you expect; you have a lot more work to do before you can start your Samba server. Click the Globals link at the top of the menu, and continue to the next section.

The Globals Menu

You can configure the `[global]` settings in your `smb.conf` file through the Globals menu. Click the link, and you should see a menu similar to the one shown in Figure 24.13.

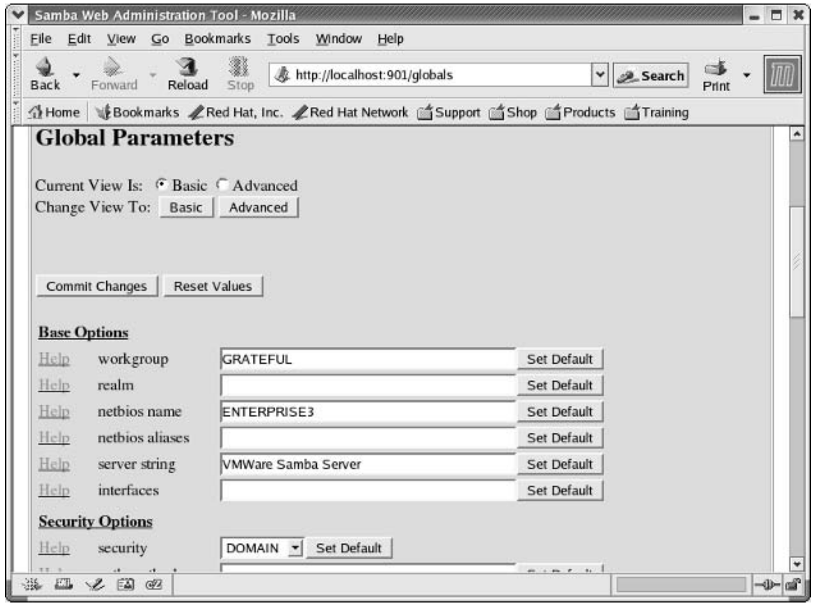


FIGURE 24.13
SWAT global variables

As you can see, a Help option is associated with each variable. Clicking Help opens the `smb.conf` man page in a new browser window, at the section with the desired variable.

In this and each of the other menus, you'll have access to the following three buttons: Commit Changes, Reset Values, and Advanced. With individual settings, you'll also see a Set Default button. Their functions are summarized in Table 24.8.

TABLE 24.9: BASIC SWAT OPTION BUTTONS

Button	Description
Commit Changes	Writes the changes you make to `smb.conf`
Reset Values	Restores the current values in `smb.conf` to the menu
Advanced view	Provides additional settings
Set Default	Activates the default setting associated with the variable

If you've read the "Samba Global Settings" section earlier in this chapter, several of the settings should look familiar to you. While I won't repeat the discussion of each variable, the way the variables are organized can help you understand how global settings work. These categories are listed in Table 24.9. Some of these categories appear only after you click the Advanced button.

TABLE 24.10: GLOBAL VARIABLE CATEGORIES

CATEGORY	DESCRIPTION
Base	Specifies the basic options for the Samba server.
Security	Allows you to configure passwords, user accounts, and computers that are allowed to connect.
Logging	Lets you customize how and where information is logged.
Protocol	Customizes interaction with different Windows protocols.
Tuning	Permits you to optimize the performance of the Samba server.
Printing	Sets the basic print protocol; the standard Linux options are `cups` and `lprng`.
Filename Handing	Allows you to set how short and regular filenames are transferred between computers.
Domain	If you're configuring this computer as a domain controller (PDC or BDC), this allows you to set administrative and guest groups.
Logon	If you're setting up this Samba server as a logon controller, this allows you to configure logon and script file locations.
Browse	Configures the priority of this computer for the Microsoft browse list of shared directories and printers.
WINS	Sets basic options for using WINS and DNS servers.
Locking	Files are locked to prevent multiple users from writing to the same file simultaneously.
LDAP	If you've set up LDAP authentication (see Chapter 23), you can use it on a Microsoft-style network.
VFS	Allows the use of the Microsoft Distributed Filesystem tree.
Winbind	Works with the `/etc/nsswitch.conf` file for resolving computer hostnames.

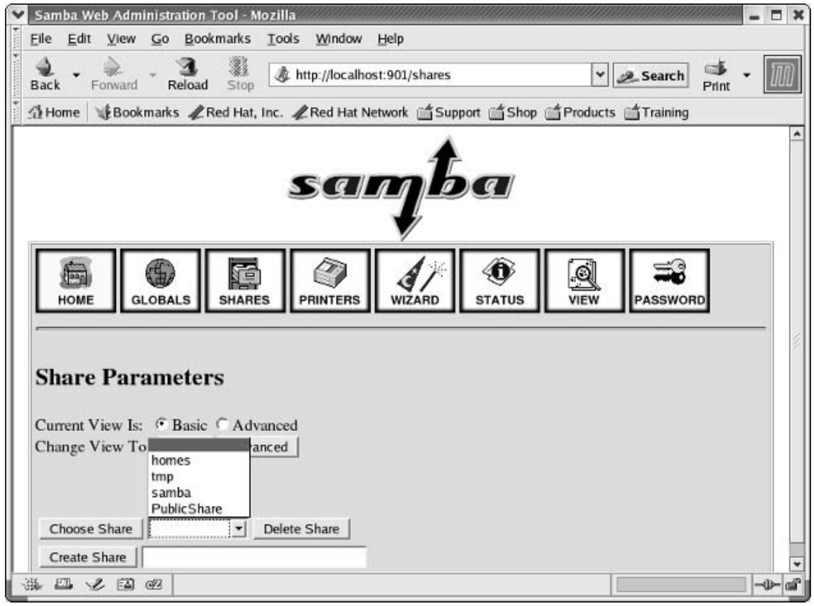
When you've completed your changes, don't forget to click the Commit Changes button to record them in `smb.conf`. Click the Shares link at the top of the menu, and continue to the next section.

TIP *Each SWAT variable includes a Help hyperlink. Click this hyperlink to open up a new browser window with more information on that variable.*

The Shares Menu

In the SWAT Shares menu, you can configure existing shares or create new ones. The initial Shares menu is shown in Figure 24.14; you need to select a share before you can customize it.

FIGURE 24.14
The SWAT Shares menu



Existing shares are taken from the names listed in the `smb.conf` file; typical shares from the packaged `smb.conf` file include `[homes]` and `[tmp]`. Select an existing share, and then click Choose Share.

Alternatively, you can configure a new shared directory. Enter the name of your choice in the Create Share text box, and then click Create Share to get to the full Shares menu.

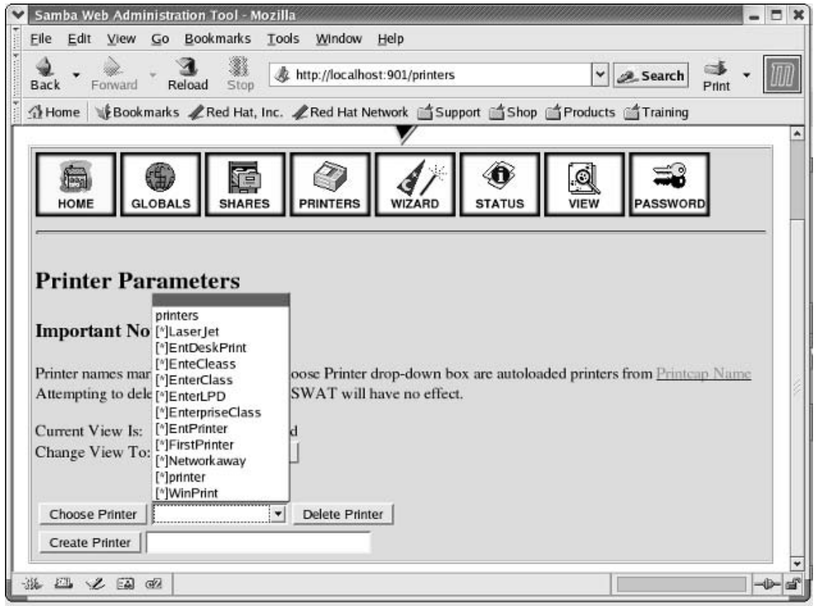
This menu illustrates the share parameters associated with the `[homes]` shared directory. We described all these variables earlier in this chapter. If you prefer, click the Advanced buttonfor more configuration options.

When you've completed your changes, don't forget to click the Commit Changes button to record them in `smb.conf`. Now click the Printers link at the top of the menu, and continue to the next section.

The Printers Menu

The SWAT Printers menu is similar to the Shares menu. You can choose to configure an existing printer, or you can create a new printer. Once you've made your selection, click the Choose Printer or Create Printer link. Individual options under the Choose Printer drop-down text box are read from `/etc/printcap`. Once you've made your selection, you can customize different printer variables, as shown in Figure 24.15.

FIGURE 24.15
SWAT Printer Parameters



When you've completed your changes, click the Commit Changes button to record them in `smb.conf`. Now click the View link at the top of the menu, and continue to the next section.

The View Menu

The SWAT View menu gives you a look at your `smb.conf` configuration file. You'll note that all of the comments and most of the settings from the original `smb.conf` file are deleted. If a variable is set to its default value, it isn't included in the normal view, similar to what's shown in Figure 24.16.

However, if you click the Full View button, you'll see an `smb.conf` file in full glory, with all available variables. You can then click the Normal View button to return to the current file. At this point, click the Password link at the top of the menu, and continue to the next section.

The Password Menu

The SWAT Password menu lets you manage the Samba passwords stored on the local Samba server, as well as manage passwords on remote computers. As you can see in Figure 24.17, this menu consists of two sections. Naturally, this can help you manage both Linux and Samba username and password databases.

Server Password Management

The Server Password Management section allows you to manage your Samba passwords, which are sent when you try to connect to a remote Samba or Microsoft Windows server. The buttons shown in Figure 24.17 are fairly self-explanatory; they are listed in Table 24.11.

FIGURE 24.16
The saved smb.conf file

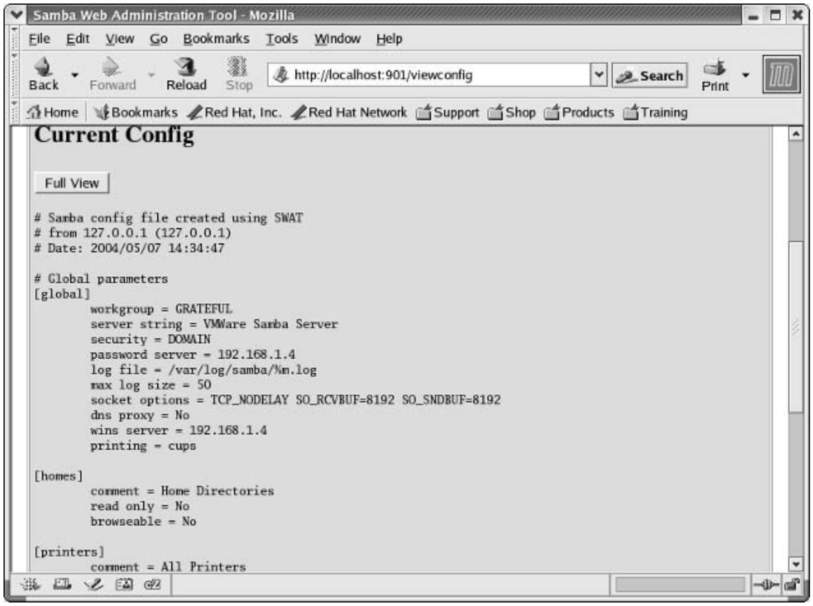


FIGURE 24.17
Managing passwords

Table 24.11: Server Password Management Functions

Function	Description
Change Password	Allows you to change the Samba password for Samba users; each user must already exist in `/etc/passwd`.
Add New User	Adds a new user to `/etc/samba/smbpasswd`; the user must already exist in `/etc/passwd`; however, this is not as flexible as the `smbadduser` command described earlier.
Delete User	Deletes the user from `/etc/samba/smbpasswd`.
Disable User	Prevents the user from connecting to remote Samba or Microsoft Windows servers.
Enable User	Allows the user to connect to remote Samba or Microsoft Windows servers.

Client/Server Password Management

This section actually allows you to change your password on a remote Microsoft Windows or Samba server. In the example shown in Figure 24.17, I changed the password for user mjang on a computer named experimental running Microsoft Windows 2000.

This won't work if Samba can't find the name of your computer in `/etc/hosts`, DNS, or possibly WINS. It also won't work if the username does not exist on the remote computer. Now click the Status link at the top of the menu, and continue to the next section.

The Server Status Menu

Once users start to connect from other computers to your Samba computer, you'll want to check your server status. As an example, Figure 24.18 shows a Samba server with connections from several other computers, with private IP addresses.

This is where SWAT is an active administration tool. At the top of the screen, you can see that the status of each connection is refreshed every 30 seconds. You can stop or restart the `smbd` and `nmbd` daemons. Under Active Connections, you can review the status of connections from different computers. You can also disconnect remote users by clicking the appropriate X button in the Kill column.

Whenever you make changes to `smb.conf`, you should reload or restart the `smbd` daemon. When you restart `smbd`, `nmbd` is restarted automatically.

Using the Red Hat Samba Server Configuration Tool

There is a simpler, though less flexible, way to configure your computer: the Samba Server Configuration tool, also known as `redhat-config-samba`. Install the `redhat-config-samba` RPM if required, and then run the program with the same name. You should see a tool similar to Figure 24.19.

This tool reads its initial settings from your current `smb.conf` file. The settings you see in Figure 24.19 reflect some of the shared directories from the packaged `smb.conf` file.

Tip *Before you open the Samba Server Configuration tool, back up your current* `smb.conf` *file.*

FIGURE 24.18
Server status

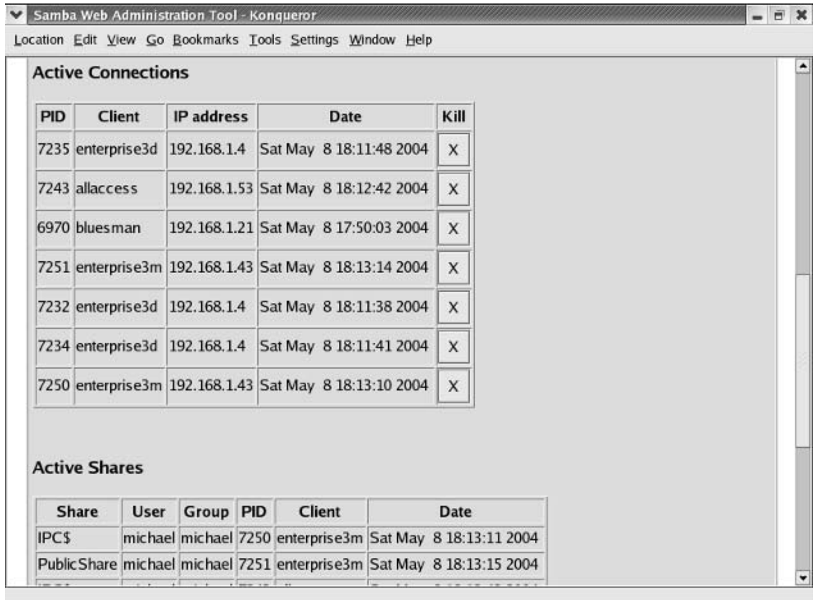
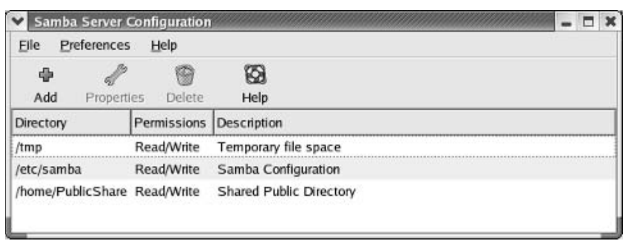


FIGURE 24.19
redhat-config-samba



NOTE *The* `redhat-config-samba` *tool is fairly new, and its features are still subject to significant changes. Use it carefully. SWAT is a more mature tool and is at least my preferred choice. However, Red Hat's SWAT packages do not include much from the* `foomatic` *package database. For more information on a version of SWAT with a full* `foomatic` *database, see* `www.cups.org` *and* `www.linuxprinting.org`.

In the following sections, we'll look at configuring basic server settings, managing Samba users, and creating a new share.

Server Settings

In `redhat-config-samba`, click Preferences ➢ Server Settings to get to the Server Settings dialog box shown in Figure 24.20. As you can see, the basic server settings are simple, and the options should be familiar from previous sections.

If the Workgroup textbox were blank, you'd know that the `workgroup` variable is set to the default, which is WORKGROUP.

FIGURE 24.20
The Basic tab of the Server Settings dialog box

Click the Security tab to see several key security variables. Remember, you can select between Active Directory (ADS), Share, User, Server, or Domain authentication modes. If you've selected ADS, Server, or Domain, you'll get to specify the name of the authentication server. Most current Windows servers use encrypted passwords. The Guest Account is an account you can designate in `/etc/passwd`, such as guest or ftp. These options are illustrated in Figure 24.21.

FIGURE 24.21
The Security tab of the Server Settings dialog box

User Management

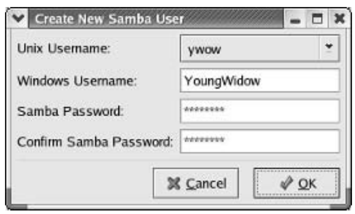
You can manage users in `redhat-config-samba`. In this tool, click Preferences ➢ Samba Users to get to the Samba Users dialog box shown in Figure 24.22.

The options are fairly self-explanatory; you can add, edit, or delete users from the Samba user database.

If you have someone with different Linux and Microsoft usernames, click Add User to see the Create New Samba User dialog box. In the text boxes shown in Figure 24.23, you can enter the Microsoft username and passwords for this person.

FIGURE 24.22
Managing Samba users

FIGURE 24.23
The Create New Samba User dialog box



Creating a New Share

You can add a new directory share with `redhat-config-samba`. In this tool, click the Add button to reveal the Create Samba Share dialog box. There are several options shown in Figure 24.24.

FIGURE 24.24
The Create Samba Share dialog box

The options are fairly straightforward, but since they don't directly match the variables described earlier, we've summarized each option under the Basic tab in Table 24.12.

TABLE 24.12: CREATING A SAMBA SHARE

OPTION	DESCRIPTION
Directory	Specifies the directory you want to share.

Continued on next page

Table 24.12: Creating a Samba Share *(continued)*

Option	Description
Browse	Calls a Select Directory dialog box to help you find the directory you want to share.
Description	Corresponds to an `smb.conf` comment.
Read-Only	Remote users aren't permitted to write to this directory.
Read/Write	Remote users are allowed to write to this directory.

Under the Access tab, you can allow access to specified users as configured or allow access to all users.

Summary

Samba is a heterogeneous service that bridges the gap between Linux and Microsoft Windows. Once you've configured Samba, you have a Linux computer that looks just like a Microsoft Windows member server on a workgroup or domain. You can even configure Samba to act just like a Microsoft PDC.

You can configure Linux as a Samba client. With the right packages, you can even use the `mount` command to connect to a shared directory from another Samba server or any Microsoft Windows server. You can even connect to a shared directory in terminal mode similar to an FTP connection.

The Samba configuration files are located in `/etc/samba`; the key file is `smb.conf`. The original `smb.conf` from the `samba-*` RPMs includes several comments that help you learn more.

The `smb.conf` file includes global settings that determine how your server connects to a Microsoft network. You can configure security, printer lists, log files, customized logon directories, browse priorities, and more. It also includes share settings, which let you configure different directories and printers. You can limit your share by user, determine how files are accessed and written, and more. Once you've configured `smb.conf`, the `testparm` command helps you check its syntax.

SWAT is a web browser–based tool for configuring `smb.conf`. Remember to activate the `swat` service in the `xinetd` daemon. It is highly customizable, with Home, Globals, Shares, Printers, View, Password, and Server Status menus.

Samba user and computer management provides the commands that allow you to manage the users and computers on your Microsoft-style network. You may have people with different Linux and Microsoft usernames on the same account. You can set up computer accounts that allow you to set up your Linux computer with Samba as a Microsoft-style PDC.

Red Hat has developed a simpler alternative to SWAT: `redhat-config-samba`. Since it is fairly new, use it with caution. While it can help you configure basic shares, it does not have the flexibility of SWAT. In my opinion, SWAT is still the preferred GUI tool for Samba.

In the next chapter, we'll examine the most important web server on the Internet, Apache. You'll learn to configure it to serve web pages on your local network and more.

Chapter 25

Web Services

THE DEVELOPMENT OF LINUX closely parallels the growth of the World Wide Web. As described in Chapter 1, Linux is based on software developed by a community of volunteers. Apache, the most popular web server in use today, was also developed by a community of many of the same volunteers. So it should not be surprising that the success of Linux is closely tied to Apache and the World Wide Web.

In 1995, the most popular web server was the HTTP daemon (HTTPd) from the National Center for Supercomputing Applications (NCSA) at the University of Illinois. When the developers of this web server left NCSA, several webmasters from around the world started updating and maintaining changes through patches, which led to its description as "a patchy" server. Thus, their web server software is known as *Apache*.

Because Apache and Linux developed in a similar way, their fortunes are closely aligned. However, Apache is also used on other Unix-style operating systems as well as Microsoft Windows. According to a Netcraft survey (`www.netcraft.com/survey`), Apache is by far the most popular web server on the Internet and has been since early 1996.

This chapter covers the version of Apache included with Red Hat Enterprise Linux 3: version 2.0.46. Later versions are available from the Apache project website, `httpd.apache.org`. However, you may want to stick with the Enterprise versions of Apache, as it incorporates the security, performance, and interoperability features of the Stronghold Enterprise web server.

In addition, Stronghold includes Red Hat's Content Accelerator, formerly known as TUX, which is a kernel-based web server designed to speed delivery of static information (such as pictures) and can be configured to work closely with Apache.

Finally, one service commonly associated with Apache is Squid, which is a caching service for frequently used content. Data associated with commonly used web pages are often stored in a Squid Proxy cache. This chapter covers the following topics:

- Exploring web server options
- Learning Apache basics
- Configuring Apache
- Configuring with the Red Hat Apache GUI tool

- Incorporating the Red Hat Content Accelerator
- Caching services

Exploring Web Server Options

Apache is not the only web server available; there are actually some proprietary web servers that you can buy. Table 25.1 briefly describes several of the important ones. According to the Netcraft survey, four web servers are currently run by more than one percent of the websites on the Internet: Apache, Microsoft's Internet Information Server, Zeus, and Sun Microsystems's Sun One.

TABLE 25.1: WEB SERVERS

NAME	DESCRIPTION
AOLServer	Used by America Online; this is an open-source web server. More information is available from `www.aolserver.com`.
Apache	The most popular web server on the Internet; more information is available from `httpd.apache.org`.
Boa	A high-performance, open-source web server that, unlike other web servers, runs most connections as a single process. More information is available from `www.boa.org`.
Caudium	A modular open-source web server. Like Boa, it runs most standard connections as a single process. More information is available from `www.caudium.net`.
Jigsaw	A web server developed by the World Wide Web Consortium (W3C). See `www.w3.org/Jigsaw` for more information. All software from this consortium conforms to their open source license.
Red Hat Content Accelerator	A kernel-based high-performance web server, formerly known as TUX. For more information, see `www.redhat.com/docs/manuals/tux`.
Resin	A server based on JavaServer Pages (JSP); more information is available from `www.caucho.com/resin`. This is a proprietary server available for purchase.
Roxen	A secure web server licensed under the GPL. For more information, see `www.roxen.com`.
Servertec	An application web server written in the Java programming language. See `www.servertec.com` for more information.
Stronghold	The security-enhanced version of Apache whose features are incorporated in the default Web server for Red Hat Enterprise Linux 3; for Red Hat documentation, see `stronghold.redhat.com`.
Sun One	A web server from Sun Microsystems; formerly known as iPlanet, it is now part of the Java System series of web servers. More information is available from `wwws.sun.com/software`.
WN	A small, secure web server, licensed under the GPL. The U.S. website is available at `hopf.math.nwu.edu`.
Zeus	A commercial high-capacity web server. More information is available from `www.zeus.co.uk`.

Learning Apache Basics

Apache is a web server. In other words, it is a service that runs on an operating system such as Linux, and it responds to requests. When users enter the address of a desired web page into a browser, their computers look to DNS servers to find the IP address of the desired web server. Once contact is made, the browser asks for the web page, usually on TCP/IP port 80. Apache responds to such requests by sending a web page to the requesting computer.

If you're currently running a web server based on Apache 1.3.*x*, you have some decisions to make. Red Hat Enterprise Linux 3 includes Apache version 2.0.46. If you install these Apache packages, you may need to make several configuration changes. You should not upgrade your Apache server until you understand and have tested your websites on the new system.

Apache 2.0

Red Hat incorporated Apache version 2.0.*x* for the first time in Red Hat Linux 8.0. Apache version 1.3.*x* is still in common use. Many of you experienced with Apache may not be familiar with the changes in version 2.0.*x*, which include the following:

- The Virtual Hosts features allow you to configure completely different websites using the same IP address.
- Directives have been changed. Those related to Perl, PHP (PHP Hypertext Processor), Python, Structured Query Language (SQL), and the Secure Sockets Layer (SSL) now have their own configuration files in the `/etc/httpd/conf.d` directory.
- Variables have changed. For example, you'll learn how to change the TCP/IP port associated with Apache using the `Listen` variable later in this chapter.
- Packages are more modular. We'll look at the different packages associated with Apache in the next section.
- Threads are used efficiently. Threads can share common data; in Apache 2.0, threads are normally processed based, which prevents server crashes. Multi-Processing Modules (MPM) support customization in this area, which helps you optimize Apache for the host operating system.
- IPv6 addresses can be used. While there's a patch that allows the use of IPv6 addresses in Apache 1.3.*x*, it is no longer recommended.

While some of these features have been "back-ported" to Apache 1.3.*x* (one reason why I think these "older" Apache servers will be around for some time), they were developed for Apache 2.0.

Stronghold Features

On top of Apache 2.0, Red Hat has incorporated the features of its Stronghold 4 web server into Enterprise Linux 3. It is also available for other Unix-type operating systems. Stronghold includes a number of features that promote security, performance, and interoperability:

- OpenSSL 0.9.7 supports 128-bit encryption using Secure Socket Layer (SSL) and Transport Layer Security (TLS).

- mod_authz_ldap allows you to use an LDAP directory to authenticate users who connect through your Apache server.
- The Red Hat Content Accelerator uses kernel-level processes to speed access to static web information.
- Cryptographic accelerators, including Rainbow, nCipher, AEP, Baltimore, and Broadcom, are also supported.
- Several different scripting languages are supported, including PHP 4.1, Perl, AxKit, mod_dav, and FrontPage 2000 server extensions.

If you install the Stronghold Apache server on a "rebuild" of Red Hat Enterprise Linux, you should monitor the associated errata. It's available online at https://rhn.redhat.com/errata/rhshas-errata.html. If you have an official Enterprise server subscription to the Red Hat network, Red Hat should notify you of these issues at least by e-mail.

Packages

Apache is a modular server. It's part of the Web Server package group; the only required package in this group is httpd-*. There are a number of other Apache packages that you can install, as shown in Table 25.2.

Table 25.2: Apache Packages

Name	Description
httpd	Installs the main Apache server
tux	Adds a kernel-based web server
bcel	Includes the Byte Code Engineering Library for managing Java class files
commons-beanutils	Supports wrappers for Jakarta and Java solutions on Apache
commons-collections	Includes additional Jakarta commons collections components
commons-digester	Installs a digester component that maps XML to Java
commons-logging	Supports logging for Jakarta
commons-modeler	Associated with Model MBeans
crypto-utils	Adds software for managing and creating SSL keys
distcache	Allows the use of caching for secure connections
distcache-devel	Adds the development libraries associated with secure caching connections
httpd-devel	Adds Apache development tools
hwcrypto	Allows interfaces with Linux hardware cryptographic accelerators
jakarta-regexp	Includes the Jakarta regular expression package

Continued on next page

TABLE 25.2: APACHE PACKAGES *(continued)*

NAME	DESCRIPTION
`mod_auth_pgsql`	Allows access limits to PostgreSQL databases
`mod_auth_mysql`	Supports access limits to MySQL-based databases
`mod_authz_ldap`	Supports authentication via LDAP directories
`mod_python`	Adds a Python language interpreter to Apache
`mod_perl`	Adds a Perl language interpreter to Apache
`mod_ssl`	Includes SSL security in Apache
`mx4j`	Adds Java Management Extensions
`php`	Installs PHP for dynamic scripts (PHP stands for PHP: Hypertext Preprocessor)
`php-imap`	Provides IMAP mail server support to Apache
`php-ldap`	Allows LDAP support for Apache
`php-mysql`	Implements PHP support of MySQL-based databases
`php-odbc`	Allows PHP interaction with Open Data Base Connectivity (ODBC)-based databases
`php-pgsql`	Installs a PHP interface with PostgreSQL-based databases
`redhat-config-httpd`	Adds the Red Hat GUI configurator for Apache
`redhat-java-rpm-scripts`	Adds a group of Java scripts for RPM packages
`squid`	Installs a proxy server
`webalizer`	Includes a log analysis program for your web server
`erces-j`	Adds an XML parser and generator
`lan-j`	Installs an XSLT processor for transforming XML into HTML

Configuring Apache

Once you've installed the desired Apache packages, your server should be ready to serve web pages to the local computer. All you need to do is start the `httpd` service and direct your web browser to the *localhost* address.

But a web server doesn't do you much good unless you can call its web pages from other computers. In this chapter, we'll analyze the main Apache configuration file, `httpd.conf`, in some detail.

These settings are based on the specifications of the Hypertext Transfer Protocol (HTTP) standards version 1.1. We provide only a brief overview of Apache 2.0; for more information, see *Linux Apache Web Server Administration*, Second Edition (Sybex, 2002).

Starting Apache

Once you've installed the Apache packages you need, starting Apache is easy. As with other services described throughout this book, all you need to do is start the applicable script from the `/etc/rc.d/init.d` directory. In this case, the following command will work nicely:

```
# apachectl start
```

Tip *Apache includes a special command for managing its service,* `apachectl`*. You can use it to* `start`*,* `stop`*, or* `restart` *the service. Unlike* `service httpd restart`*, an* `apachectl graceful` *command restarts the service without disconnecting users.*

If you still have the default Apache configuration file, you'll probably see the following message:

```
Starting httpd: httpd: Could not determine the server's fully qualified domain
➥ name, using 127.0.0.1 for ServerName
```

Now you can open the browser of your choice to the localhost address. This is also known as the *loop-back IP address*, which, as defined in Chapter 15, is 127.0.0.1. Figure 25.1 shows the result in the Mozilla web browser.

You'll also want to use a command such as `chkconfig`, as described in Chapter 13, to make sure Apache starts the next time you start Linux at an appropriate runlevel. For example, the following command starts the Apache daemon, `httpd`, whenever you start Linux in runlevel 2, 3, or 5:

```
# chkconfig --level 235 httpd on
```

Now you're ready to start customizing the Apache configuration.

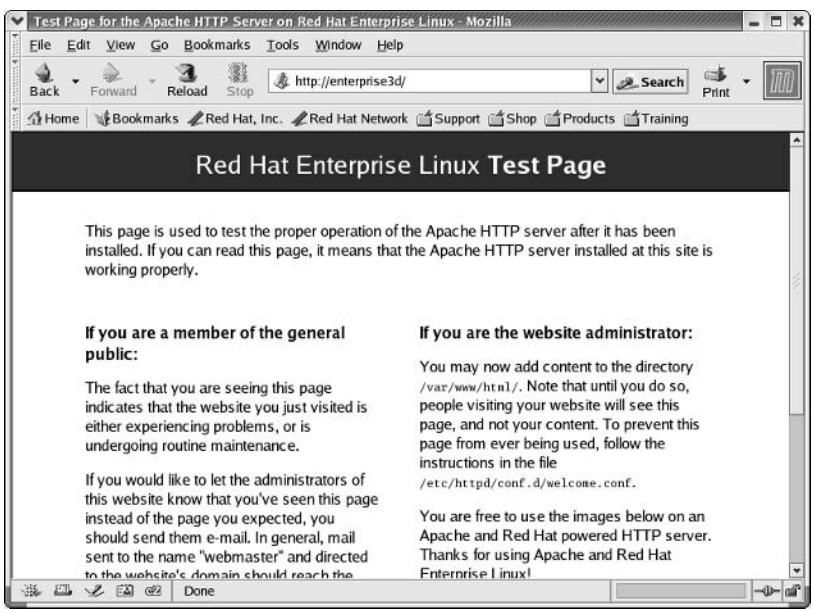


Figure 25.1
Apache is properly installed.

Customizing Apache

The main Apache configuration file, `httpd.conf`, is located in the `/etc/httpd/conf` directory. It is split into three sections. In the global environment section, you can configure the basic settings for this web server. In the main server configuration section, you'll set up the basic defaults for any websites on your server. The Virtual Hosts section allows you to set up several different websites on your Apache server, even if you have only one IP address.

NOTE *There were originally three main configuration files for Apache:* `access.conf`*,* `srm.conf`*, and* `httpd.conf`*, all located in the same directory. While later versions of Apache 1.3.*x *incorporated the information from* `access.conf` *and* `srm.conf` *in* `httpd.conf`*, at least blank versions of* `access.conf` *and* `srm.conf` *were still required by the server. Apache 2.0.*x *no longer needs these extra configuration files.*

Commands in the Apache configuration file are known as *directives*. In the following sections, we'll analyze the directives from the default Apache `httpd.conf` installed with Red Hat Enterprise Linux 3 in some detail. You can read the file for yourself; it includes many other useful comments.

Commands with a pound sign (`#`) in front are commented out in the default Apache configuration file. If you're learning about Apache for the first time, experiment a bit. Set up some website files on your computer. Use the directory specified by the `DocumentRoot` directive, which is by default `/var/www/html`. Try some of these commands, restart the `httpd` daemon, and examine the changes for yourself. You may be surprised at what you can do.

Global Environment

We'll look at each of the directives in the global environment section of the default version of the Apache `httpd.conf` configuration file. Variables in this section apply to all Virtual Hosts that you might configure on this server. There are basic parameters, detailed parameters related to different clients, port settings, pointers to other configuration files, and module locations.

NOTE *If a directive is set to 0, it normally means you're setting no limit on that directive. For example, if you set* `Timeout` *to 0, connections from a client browser are kept open indefinitely.*

Basic Global Environment Parameters

The following directive gives users of your website some basic information about your software. While the following command tells users that your web server is Apache on a Unix-style system, other commands are possible, as described in Table 25.3:

```
ServerTokens OS
```

TABLE 25.3: *SERVERTOKENS* DIRECTIVE OPTIONS

DIRECTIVE	DESCRIPTION
`ServerTokens Prod`	Identifies the web server as Apache
`ServerTokens Min`	Identifies Apache and its version number

Continued on next page

Table 25.3: *ServerTokens* Directive Options *(continued)*

Directive	Description
`ServerTokens OS`	Identifies Apache, its version number, and the type of operating system
`ServerTokens Full`	Identifies Apache, its version number, the type of operating system, and compiled modules

The `ServerRoot` directive identifies the directory with configuration, error, and log files,

```
ServerRoot "/etc/httpd"
```

If you run `ls -l /etc/httpd`, you'll see links to the real location of certain directories; for example, `/etc/httpd/logs` is linked to the `/var/log/httpd` directory.

Apache includes parent and child processes for different connections. The `ScoreBoardFile` parameter helps these processes communicate with each other. Otherwise, the communication is through active memory.

```
#ScoreBoardFile run/httpd.scoreboard
```

Tip *I normally avoid activating the* `ScoreBoardFile` *parameter; it's required only for certain architectures, and does not apply for operating systems that include memory mapping functions, including Red Hat Enterprise Linux 3.*

You may note that `run` is a relative subdirectory. The full directory name is based on the `ServerRoot` directive—in other words, `/etc/httpd/run`.

The `PidFile` specifies the file where Apache records the process identifier (PID):

```
PidFile run/httpd.pid
```

If computers are having trouble communicating on your network, you need a `Timeout` value to keep Apache from hanging. The `Timeout` directive specifies a stop value in seconds.

```
Timeout 300
```

Normally, multiple requests are allowed through each connection. The following command disables this behavior:

```
KeepAlive Off
```

If the `KeepAlive` directive is on, you can regulate the number of requests per connection with the `MaxKeepAliveRequests` directive:

```
MaxKeepAliveRequests 100
```

Once a connection is made between Apache and someone's web browser, the `KeepAliveTimeout` directive specifies the number of seconds to wait for the next client request.

```
KeepAliveTimeout 15
```

Detailed Client Parameters

Apache includes a number of Multi-Processing Modules (MPM). These MPMs fall into three categories.

- Prefork MPMs are suited to process-based web servers; they are appropriate to use if you have Apache modules that do not require separate threads. This imitates the behavior of Apache 1.3.*x*.
- Worker MPMs support both types of modules; however, they should not be used if you're using Apache 1.3 modules, since threads can cause problems.
- Per-child MPMs support websites for clients that need different user IDs.

NOTE *MPMs flexible; specific modules are available for Windows NT (*`mpm_winnt`*) and Novell Netware (*`mpm_netware`*) networks.*

There are a number of common directives that you can specify in each of thenoted MPM categories.

When Apache is started, the `StartServers` directive sets the number of available child server processes ready for users who want your web pages:

```
StartServers 8
```

Once Apache is started, requests from other users may come in. If the number of unused server processes falls below the `MinSpareServers` directive, additional `httpd` processes (also known as child servers) are started automatically.

```
MinSpareServers 5
```

When traffic goes down, the `MaxSpareServers` directive determines the maximum number of `httpd` processes (also known as child servers) that are allowed to run idle.

```
MaxSpareServers 20
```

You can regulate the number of clients requesting information from your web server with the `MaxClients` directive.

```
MaxClients 150
```

You can also regulate the number of requests for information from each client with the `MaxRequests-PerChild` directive. This is the number of requests that an individual child server will handle during its lifetime.

```
MaxRequestsPerChild 1000
```

Apache 2.0 servers can start new threads for each request. The `MinSpareThreads` directive is similar to `MinSpareServers`; it allows Apache to handle a surge of additional requests.

```
MinSpareThreads 25
```

When the number of requests goes down, Apache monitors the number of spare threads; if the number exceeds the `MaxSpareThreads` directive, some threads are killed.

```
MaxSpareThreads 75
```

Every child process can create several threads to handle requests from each user of your website. The `ThreadsPerChild` directive is created when each child process starts.

```
ThreadsPerChild 25
```

You can limit the number of threads allowed for each child process with the `MaxRequestsPerChild` directive (there is no limit in the default `httpd.conf` file):

```
MaxRequestsPerChild 0
```

You can also limit the number of threads allowed for each child process with the `MaxThreadsPerChild` directive.

```
MaxThreadsPerChild 20
```

Port Settings

You can set Apache to `Listen` to requests from only certain IP addresses and or TCP/IP ports. The default `httpd.conf` file includes the following directives:

```
Listen 0.0.0.0:80
```

If you have more than one network adapter, you can also limit Apache to certain networks; for example, the following directive listens only to the network adapter with an IP address of 192.168.13.64 on TCP/IP port 80:

```
Listen 192.168.13.64:80
```

NOTE *The* `Listen` *directive supersedes the* `BindAddress` *and* `Port` *directives from Apache version 1.3.x.*

Module Locations

When you need a module in Apache, it should be loaded in the `httpd.conf` configuration file. Normally, modules are listed in the following format:

```
LoadModule module_type location
```

For example, the following directive loads the module named `access_module` from the `ServerRoot` modules subdirectory, `/etc/httpd/modules`. You will find that this is linked to the actual directory with Apache modules: `/usr/lib/httpd/modules`.

```
LoadModule access_module modules/mod_access.so
```

Several modules are listed in the default `httpd.conf` file; Table 25.4 offers a brief description. The modules are listed in the same order as they appear in the default file.

One of the more interesting modules is `info_module`; as you'll see toward the end of the next section, it supports a detailed view of your Apache server configuration in your browser at `localhost/server-info`.

TABLE 25.4: STANDARD APACHE MODULES

MODULE	DESCRIPTION
`access_module`	Supports access control based on an identifier, such as a computer name or IP address
`auth_module`	Allows authentication (usernames and passwords) with text files
`auth_anon_module`	Lets users have anonymous access to areas that require authentication
`auth_dbm_module`	Supports authentication with DBM (database management) files
`auth_digest_module`	Sets authentication with MD5 digests
`include_module`	Includes SSI (server-side includes) data for dynamic web pages
`log_config_module`	Sets logging of requests to the server
`env_module`	Allows control of the environment that is passed to CGI (Common Gateway Interface) scripts and SSI pages
`mime_magic_module`	Sets Apache to define the file type from a look at the first few bytes of the contents
`cern_meta_module`	Supports additional meta-information with a web page, per the standards of the W3C, which is housed at CERN (the French acronym for the European Laboratory for Particle Physics)
`expires_module`	Lets Apache set an expiration date for the page, to support a web browser refresh request
`deflate_module`	Compresses content
`headers_module`	Allows control of HTTP request and response headers
`usertrack_module`	Supports user tracking with cookies
`unique_id_module`	Sets a unique identifier for each request
`setenvif_module`	Allows Apache to set environment variables based on request characteristics, such as the type of web browser
`mime_module`	Associates the filename extension, such as `.txt`, with specific applications
`dav_module`	Supports web-based distributed authoring and versioning functionality
`status_module`	Gives information on server performance and activity
`autoindex_module`	Allows the listing of files in a web directory
`asis_module`	Sends files without adding extra headers
`info_module`	Supports user access to server configuration information
`dav_fs_module`	Supports `dav_module`
`vhost_alias_module`	Allows dynamically configured Virtual Hosts

Continued on next page

TABLE 25.4: STANDARD APACHE MODULES *(continued)*

MODULE	DESCRIPTION
`negotiation_module`	Sets Apache to match content, such as language, to the settings from the browser
`dir_module`	Supports viewing of files in Apache directories
`imap_module`	Configures image map file directives (not related to e-mail)
`actions_module`	Lets you run CGS scripts
`speling_module`	Allows for small mistakes in requested document names (ironically, the module name is misspelled)
`userdir_module`	Supports access to user-specific directories
`alias_module`	Sets up redirected URLs
`rewrite_module`	Supports rewriting of URLs
`proxy_module`	Sets up a proxy server for Apache
`proxy_ftp_module`	Allows proxy server support for FTP data
`proxy_http_module`	Allows proxy server support for HTTP data
`proxy_connect_module`	Required for proxy server connect requests
`cache_module`	Stores (caches) content associated with URLs
`suexec_module`	Allows CGI scripts to be run as a specific user/group
`disk_cache_module`	Storage (cache) manager associated with URLs
`file_cache_module`	Stores a static list of files in cache
`mem_cache_module`	Stores content associated with URLs
`cgi_module`	Configures the running of CGI scripts

Pointers to Other Configuration Files

As we noted earlier, there are other configuration files associated with the Apache 2.0.*x* server. By default, they're in the `/etc/httpd/conf.d` directory. Normally, file locations are determined by the `ServerRoot` directive, which is set to `/etc/httpd`, and the `Include` directive shown here:

```
Include conf.d/*.conf
```

Status and Extended Status

By default, Apache on Red Hat Enterprise Linux 3 includes the `mod_status` module. You can enable additional status logging information by activating the following command:

```
# ExtendedStatus On
```

Main Server Configuration

Before we move onto configuring Virtual Hosts, let's take a look at the next section in the `httpd.conf` configuration file, which includes the default directives for Apache. While you can configure different settings for many of these directives, you need to know the defaults in this section. We analyze the basic settings in this part of the `httpd.conf` file in order.

NOTE *This is a long section; you may want to take a break if you're in the habit of reading through a complete section at a time.*

System User

As determined by the User and Group directives, the Apache daemon, `httpd`, is assigned a specific user and group name here and in `/etc/passwd` and `/etc/group`.

```
User apache
Group apache
```

Administrative Contact

With web pages generated by Apache, there is a listing for an administrative contact, as determined by the `ServerAdmin` directive. Naturally, you can change this to the e-mail address of your choice.

```
ServerAdmin root@localhost
```

Web Server Name

If you have an administrative website for your web server, you'll want to set it with the `ServerName` directive. If you don't have a fully qualified domain name in a DNS server, or your DNS server is not completely reliable, use the IP address.

```
#ServerName new.host.name:80
```

If you activate this directive, it will normally be superseded by the name you set for each Virtual Host.

Canonical Name

Technically, every URL, such as `http://www.Sybex.com/`, is supposed to have a trailing slash. But I never remember to put it in. Without the following directive, an attempt to navigate to `www.Sybex.com` would end up at the address specified by the `ServerName` directive. The standard `httpd.conf` file includes the following `UseCanonicalName` directive to add the trailing slash automatically:

```
UseCanonicalName Off
```

Document Root

The root directory for your web server is specified by the `DocumentRoot` directive. This is where you'll write the actual HTML pages for your websites.

```
DocumentRoot "/var/www/html"
```

Web Directory Permissions

Next, we look at the default permissions for users within directories accessible through your server's websites. They're set up by the `<Directory />` container, which defines the permissions associated with the `DocumentRoot`.

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

The `Options` directive determines where you can go for files from that directory. It can be set to several different values, as described in Table 25.5. The `AllowOverride` directive can go to the `.htaccess` file for a list of users or computers allowed to see certain files; the `AllowOverride None` setting doesn't even look at the `.htaccess` file.

TABLE 25.5: ***OPTIONS*** **DIRECTIVE VALUES**

VALUE	DESCRIPTION
`All`	Supports all settings except `MultiViews`.
`ExecCGI`	Allows the running of CGI scripts.
`FollowSymLinks`	Lets requests follow symbolically linked files or directories.
`Includes`	Allows the use of server-side includes (SSI).
`IncludesNOEXEC`	Allows SSIs but not CGIs.
`Indexes`	If there is no `index.html` type file, sets up Apache to return a list of files in that directory. Options for this file are specified by the `DirectoryIndex` directive.
`MultiViews`	Supports content negotiation, such as between web pages in different languages.
`SymLinksIfOwnerMatch`	Follows symbolic links if the target file or directory is owned by the same user.

Specific Directory Permissions

Next, we'll look at the default permissions in `httpd.conf` for the `DocumentRoot` directory, which is normally `/var/www/html`, as specified by the following container:

```
<Directory "/var/www/html">
```

The following `Options` directive supports redirection via symbolic links and the listing of files in the current directory if there is no `index.html` type file (look ahead to Figure 25.2 for an example):

```
Options Indexes FollowSymLinks
```

As we mentioned in the previous section, the `AllowOverride` directive specifies the types of directives in the `.htaccess` file. For other `AllowOverride` options, look ahead to Table 25.6. The following option doesn't even look at `.htaccess`:

```
AllowOverride None
```

.HTACCESS FILES

An `.htaccess` file is a distributed configuration file that you can use to configure individual directories on a website. It is a common way to implement restricted access to a specific directory.

An `.htaccess` file isn't necessary in most cases; you can configure access on a per-directory basis in the main Apache configuration file, `httpd.conf`. In the default version of the main Apache configuration file, look for `<Directory>` containers. Observe how the restrictions vary for different directories.

However, if you have a large number of websites on your server, such as the personal web pages associated with many ISPs, you may want to use `.htaccess` files to let individual users regulate access to web pages in their home directories. You can set up a standard scheme to read `.htaccess` files, as described later in the "User Directory Permissions" section.

If you want to implement distributed configuration files, you can do something to make it more secure. Look for the `AccessFileName` directive in `httpd.conf`. Assign a hidden filename other than `.htaccess`. Also see the "Access Control" section later in this chapter.

Finally, there are access control directives; the following looks for an `Allow` and then a `Deny` directive for this directory, in order:

```
Order allow,deny
Allow from all
```

User Directory Permissions

You can set up web pages in your users' home directories. They are disabled by default with the following command:

```
UserDir disable
```

You can replace that command with the following:

```
UserDir public_html
```

Assume you have a user named ez, and she has a set of web page files in the `/home/ez/public_html` directory. Also, assume that your website is named `www.example.abc`. You need to set the appropriate permissions.

```
# chmod 711 /home/ez
# chmod 755 /home/ez/public_html
# chmod 744 /home/ez/public_html/*
```

Then when you direct your browser to `www.example.abc/~ez`, you will be able to see any `index.html` web page that you may have stored in the `/home/ez/public_html` directory.

You can further regulate access to web pages and files in users' home directories. Look at the following sample commands from the default `httpd.conf` file:

```
#<Directory /home/*/public_html>
#    AllowOverride FileInfo AuthConfig Limit
```

```
#    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
#    <Limit GET POST OPTIONS>
#        Order allow,deny
#        Allow from all
#    </Limit>
#    <LimitExcept GET POST OPTIONS>
#        Order deny,allow
#        Deny from all
#    </LimitExcept>
#</Directory>
```

If you activate these commands, Apache allows you to browse the files in user `public_html` subdirectories, as described later in the "Directory Listings" section.

As described earlier, the `AllowOverride` directive relates to the access information that Apache reads from an individual `.htaccess` file. The different parameters associated with this directive are shown in Table 25.6. All descriptions refer to the commands that you can use in an `.htaccess` file on a per-directory basis.

TABLE 25.6: ***ALLOWOVERRIDE*** **DIRECTIVE PARAMETERS**

Parameter	Description
`AuthConfig`	Supports the use of authorization directives
`FileInfo`	Lets you configure various document types
`Indexes`	Permits you to configure indexing of the directory
`Limit`	Supports access control restrictions, such as `deny` and `allow`

The `Options` directive described in Table 25.5 supports content negotiation, file indexing, following symbolic links, and support for SSIs but not CGIs.

The `Limit` directive sets options for users who want to send (`POST`) and receive (`GET`) files from the user home directory; the `LimitExcept` directive denies the use of all other access commands.

Directory Index

When users navigate to your website, they're actually looking in a directory. The `DirectoryIndex` directive tells Apache the types of web pages to send back to the website user.

```
DirectoryIndex index.html index.html.var
```

The `index.html` document is a standard home page file used by many websites; `index.html.var` is one way to set up a dynamic home page. You can look at an example of `.var` files in the `/var/www/error` directory. Open those files in the text editor of your choice. You'll see standard error messages.

Access Control

As described in the sidebar "`.htaccess` Files," you can configure access control files on individual directories. By default, it's the hidden file `.htaccess`; you can set a different filename with the `AccessFileName` directive:

```
AccessFileName .htaccess
```

The following `Files` directive ensures that any file that starts with `.ht` is not viewable by users who are browsing your website:

```
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

MIME Types

While the MIME (Multipurpose Internet Mail Extensions) standard was originally created for sending binary files over e-mail, it works for web pages as well. For example, you can configure your browser to open the PDF reader of your choice if you navigate to a PDF file on the Internet. The standard translation between MIME types and file extensions is listed through the `TypesConfig` directive.

```
TypesConfig /etc/mime.types
```

Many files do not have extensions such as `.pdf` or `.doc`. You can set the `DefaultType` directive to specify display options on a browser. If you use text files, the following standard should work well:

```
DefaultType text/plain
```

Alternatively, if most of your files are in binary format, you could end up sending dozens of pages of gibberish to your users unless you changed this directive to something like:

```
DefaultType application/octet-stream
```

If the extension doesn't provide a clue, you can use the `MIMEMagicFile` directive, which uses the `mod_mime_magic` module defined in Table 25.4.

```
<IfModule mod_mime_magic.c>
#   MIMEMagicFile /usr/share/magic.mime
    MIMEMagicFile conf/magic
</IfModule>
```

Remember, the location of a "relative" path such as `conf/magic` is based on the `ServerRoot` directive. In other words, this section points to `MIMEMagicFile` at `/etc/httpd/conf/magic`.

There is one more related directive, toward the end of this section of the `httpd.conf` file. The `AddType` directive allows you to override the configuration as defined by `TypesConfig` in `/etc/mime.types`:

```
AddType application/x-tar .tgz
```

Log Data

Apache logs can be very large. If you're running a large commercial website, you could easily collect hundreds of megabytes of log data every day. The choices you make for log data could easily overload your system.

Normally, `HostnameLookups` are set to `Off`; otherwise, Apache will look for the fully qualified domain name of every requesting user. Don't do this unless you have reliable access to a DNS server and the network capacity to handle that volume of information.

```
HostnameLookups Off
```

You can set the locations of different log files. The `ErrorLog` directive, as you'd expect, sets the location of the `error_log` file. With the given value of `ServerRoot`, the following log file is located in the `/etc/httpd/logs` directory:

```
ErrorLog logs/error_log
```

You can control the types of messages sent to the `ErrorLog` file; available values for the `LogLevel` directive (`debug`, `info`, `notice`, `warn`, `error`, `crit`, `alert`, `emerg`) are similar to those shown in the standard error log file, `/etc/syslog.conf`, in Chapter 13.

```
LogLevel warn
```

Log information is sent to the `error_log` in a specific format, as defined by the following `LogFormat` directives:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

Each of these lines specifies a set of data collected in four different formats: `combined`, `common`, `referer`, and `agent`.

The variables associated with `LogFormat` are described in Table 25.7. A substantial number of additional variables are available, which you can review in the `mod_log_config.html` file in the `/var/www/manual/mod` directory. Other request fields are per the standards of the World Wide Web Consortium, at `www.w3.org/Protocols/HTTP/HTRQ_Headers.html`.

TABLE 25.7: *LOGFORMAT* DIRECTIVE VARIABLES

VARIABLE	DESCRIPTION
%a	Remote IP address.
%b	Bytes sent (not including HTTP headers).
%h	Remote host.
%l	Remote log name.
%r	First line of the client request.

Continued on next page

TABLE 25.7: *LOGFORMAT* **DIRECTIVE VARIABLES** *(continued)*

VARIABLE	DESCRIPTION
`%s`	Request status.
`%t`	Time.
`%u`	Remote user.
`referer`	Notes the page where someone clicked a link. (Yes, in Apache, *referer* is not spelled correctly.)
`user-agent`	Notes the client program, such as Mozilla.

You can set the location of several other types of logs, as defined through the `CustomLog` variable. You can set this up within one of your Virtual Hosts, so the owners of individual websites on your server can get their own log files.

```
# CustomLog logs/access_log common
CustomLog logs/access_log combined
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
#CustomLog logs/access_log combined
```

These lines specify the location of your log files. Based on the default `ServerRoot`, that's `/etc/httpd/logs`. The actual information that's sent to each log file is based on the referenced `LogFormat`. For example, the active `CustomLog` directive refers to the `combined` format, which you might recall is:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

Memory Mapping of Files

Memory mapping of Web-related files is enabled by default. This often results in better performance. However, you should activate the following directive if you're also sharing the `DocumentRoot` directory using NFS. As memory mapping can be a problem for multi-CPU systems, some trial and error may be appropriate.

```
#EnableMMAP off
```

Until recently (before Apache version 2.0.44), Apache actually read the content of Web files before delivering them to clients. But this isn't necessary for Web files with static content. By default, Apache now just delivers static Web files to client browsers. But this can be troublesome if your `DocumentRoot` is shared using Samba or NFS. It can trigger TCP-checksum related bugs when some network cards are configured with IPv6 addresses. If you have either of these issues on your Web server, you may want to activate the following command:

```
#EnableSendfile off
```

The Server Signs the Web Page

The `httpd.conf` file can add one element to dynamically generated web pages, depending on the `ServerSignature` directive. Normally it's set as follows:

```
ServerSignature On
```

When `ServerSignature` is set to `On`, you may see a message similar to the following at the bottom of dynamically generated web pages:

```
Apache/2.0.46 (Red Hat) Server at localhost Port 80
```

Alternatively, if you substitute `Email` for `On`, you'll get a hyperlink from the name of the computer, in this case, `localhost`, to the server administrator, as defined by the `ServerAdmin` directive.

Aliases

You can use the `Alias` directive to set up a link between a directory in the URL to a directory on your computer. For example, the first `Alias` directive in the default `httpd.conf` file links the `/icons/` subdirectory from a URL

```
Alias /icons/ "/var/www/icons/"
```

to the `/var/www/icons/` directory on the web server. This is also a good place to specify the permissions associated with `/var/www/icons/`.

```
<Directory "/var/www/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

These permissions allow users to read the contents of the directory, unless there's a `DirectoryIndex` file such as `index.html`, and support content negotiation, such as different languages, via `MultiViews`.

If you've installed a third-party version of the `httpd-manual-*` RPM and want to include the Apache manual on your website, change the following default `Alias` directive from

```
Alias /manual "/var/www/manual"
```

to

```
Alias /etc/httpd/manual "/var/www/manual"
```

This assumes that your `ServerRoot` directive is set to `/etc/httpd`. The following lines set permissions for the noted directory and include the Web-based Distributed Authoring and Versioning (WebDAV) database:

```
<Directory "/var/www/manual">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
```

```
    Allow from all
</Directory>

<IfModule mod_dav_fs.c>
    # Location of the WebDAV lock database.
    DAVLockDB /var/lib/dav/lockdb
</IfModule>
```

NOTE *Red Hat Linux 9 included an* `httpd-manual` *RPM for version 2.0.40. While you can install this older RPM, we prefer the latest information, available online at* `httpd.apache.org/docs-2.0`*.*

Scripts

Scripts in `httpd.conf` refer to programs that are run through the web server. Apache starts in the default `httpd.conf` file with a `ScriptAlias` directive, which is a specialized `Alias` for scripts:

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

Some scripts require access to the CGI daemon, which is defined by the `Scriptsock` directive. If this describes your situation, you may want to add the following directives:

```
<IfModule mod_cgid.c>
    Scriptsock    run/httpd.cgid
</IfModule>
```

Once again, this is a good opportunity to define the permissions associated with the scripts associated with your websites.

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

Note how these permissions don't allow the use of `.htaccess` but support script execution by all users.

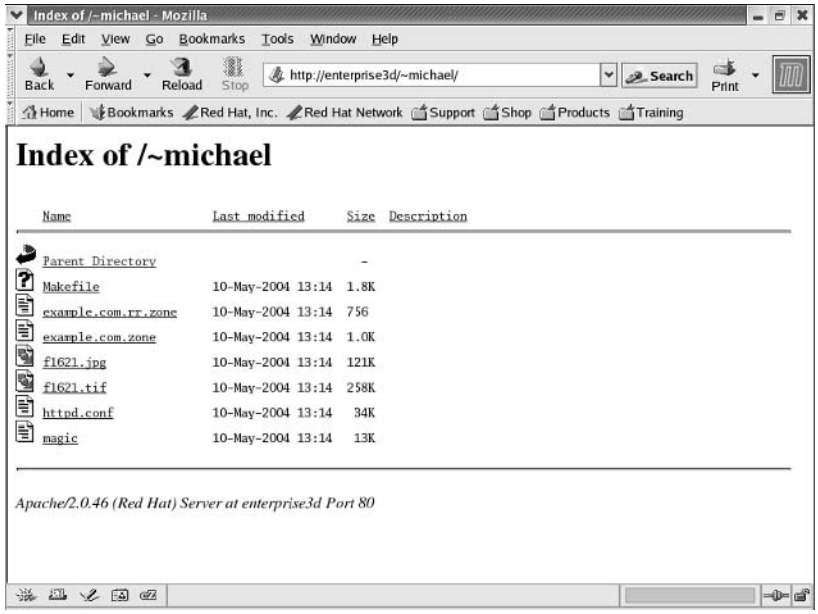
If you change website names, you'll want to redirect users. For example, the following default `Redirect` directive takes users who navigate to your `/bears` directory to `www.mommabears.com`:

```
# Redirect permanent /bears http://www.mommabears.com
```

Directory Listings

Sometimes you want to see the files in a directory. For example, Figure 25.2 illustrates the files in the `/home/michael/public_html` directory, based on the `UserDir` directives described earlier in the "User Directory Permissions" section.

FIGURE 25.2
Viewing home directory files



The `IndexOptions` directive determines how index files are shown in client web browsers. For example, the default `IndexOptions` line

```
IndexOptions FancyIndexing VersionSort NameWidth=*
```

configures `FancyIndexing`, for icons and file sizes; `VersionSort`, which sorts numbers such as RPM versions in a specific order; and a `NameWidth` as large as needed for the filenames in the directory.

Icons

Speaking of icons, a list of icons is available for different file types and extensions. These icons are shown with a file list, assuming you have set `IndexOptions FancyIndexing` as defined in the previous section. There are three basic `AddIcon*` directives:

```
AddIconByEncoding (CMP,/icons/compressed.gif)  x-compress x-gzip
```

The `AddIconByEncoding` directive shown here applies to compressed binary files. Several `AddIconByType` directives are also included for four different file types.

```
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
```

Finally, there are a series of `AddIcon` directives that associate a specific icon with different filename extensions:

```
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
```

These `AddIcon` directives are straightforward. For example, if Apache sees a file with an `.exe` extension, it adds the `/icons/binary.gif` icon as a label for that particular file. But this list is not comprehensive; there is a `DefaultIcon` directive for files with unknown extensions:

```
DefaultIcon /icons/unknown.gif
```

If you like, you can activate the following `AddDescription` directives to give users a bit more information about files with specific extensions:

```
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz
```

You can set up directories with various HTML files. For example, the `HeaderName` directive specifies a file to put before the file list; the `ReadmeName` directive specifies a file to put after the file list.

```
ReadmeName README.html
HeaderName HEADER.html
```

The `IndexIgnore` directive sets Apache to avoid listing the noted files in any directory list. Note how the default value includes the `HEADER.html` and `README.html` files.

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

Decompression

Some browsers can read and automatically decompress certain files in your website directories. All you need to do is specify the encoding associated with certain filename extensions by using the `AddEncoding` directive:

```
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz
```

Languages

Multilingual websites include web pages in multiple languages. The `DefaultLanguage` directive defines the language associated with all web pages that aren't already labeled. The following inactive directive specifies the Dutch language:

```
# DefaultLanguage nl
```

You can set up web pages in different languages, as defined by the `AddLanguage` directive. For example, `index.html.ru` is a web page associated with the Russian language.

```
AddLanguage ru .ru
```

Other language codes are listed in Table 25.8.

TABLE 25.8: ***AddLanguage*** **CODES**

CODE	LANGUAGE
ca	Catalan
cs	Czech
da	Danish
de	German
en	English
el	Modern Greek
es	Spanish
et	Estonian
fr	French
he	Hebrew
hr	Hungarian
it	Italian
ja	Japanese
ko	Korean

Continued on next page

TABLE 25.8: *ADDLANGUAGE* **CODES** *(continued)*

CODE	LANGUAGE
`ltz`	Luxembourgeois
`nl`	Dutch (Netherlands)
`nn`	Norwegian Nynorsk
`no`	Norwegian
`pl`	Polish
`pt`	Portuguese
`pt-br`	Brazilian Portuguese
`ru`	Russian
`sv`	Swedish
`zh-cn`	Chinese *
`zh-tw`	Chinese

There have been recent changes for several languages; Korean used to be `kr`*, Czech was* `cz`*, and Chinese was* `tw`*. You'll still see the old designations as part of the character set names.*

A web browser should tell the web server the preferred language. However, when this doesn't work, the `LanguagePriority` directive sets the preferred language.

```
LanguagePriority en da nl et fr de el it ja ko no pl pt pt-br ltz ca es sv zh-cn
```

NOTE *There is an error in the* `LanguagePriority` *directive in the default file. The last entry in the default list is* `tw`*, which is now obsolete.*

This works hand in hand with the `ForceLanguagePriority` directive. As defined in the default `httpd.conf` file, it uses the `LanguagePriority` directive list to select from languages acceptable to the client web browser. If no acceptable language page is available, the first item on the `LanguagePriority` list (in this case, English) is used.

```
ForceLanguagePriority Prefer Fallback
```

Many languages don't work too well unless you've installed the right set of characters. Most language characters have been organized into different ISO character sets. The default, which translates the English alphabet into binary code, is UTF-8. It's forced into the default websites for Apache with the following directive:

```
AddDefaultCharset UTF-8
```

NOTE *Incidentally, Ken Thompson gets credit as the developer of UTF-8. With Dennis Ritchie, Thompson is credited as the original developer of Unix.*

Several other character sets are available, as defined by the following `AddCharset` directives. For more information on these character sets, see `www.iana.org/assignments/character-sets`.

```
AddCharset ISO-8859-1  .iso8859-1  .latin1
AddCharset ISO-8859-2  .iso8859-2  .latin2 .cen
AddCharset ISO-8859-3  .iso8859-3  .latin3
AddCharset ISO-8859-4  .iso8859-4  .latin4
AddCharset ISO-8859-5  .iso8859-5  .latin5 .cyr .iso-ru
AddCharset ISO-8859-6  .iso8859-6  .latin6 .arb
AddCharset ISO-8859-7  .iso8859-7  .latin7 .grk
AddCharset ISO-8859-8  .iso8859-8  .latin8 .heb
AddCharset ISO-8859-9  .iso8859-9  .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5        .Big5       .big5
# For Russian, more than one charset is used (depends on client, mostly):
AddCharset WINDOWS-1251 .cp-1251   .win-1251
AddCharset CP866       .cp866
AddCharset KOI8-r      .koi8-r .koi8-ru
AddCharset KOI8-ru     .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8       .utf8
AddCharset GB2312      .gb2312 .gb
AddCharset utf-7       .utf7
AddCharset utf-8       .utf8
AddCharset big5        .big5 .b5
AddCharset EUC-TW      .euc-tw
AddCharset EUC-JP      .euc-jp
AddCharset EUC-KR      .euc-kr
AddCharset shift_jis   .sjis
```

Mapped Handlers

You can map filename extensions to a specific handler. For example, the following commented `AddHandler` directive activates CGI script handling for files with the `.cgi` extension, assuming you also have set the `Options ExecCGI` directive for the subject directory:

```
#AddHandler cgi-script .cgi
```

The following commented directive makes sure that files that already have HTTP headers don't get processed:

```
#AddHandler send-as-is asis
```

To activate commented directives, remove the comment mark (`#`) in `httpd.conf` in the text editor of your choice.

This directive processes image map files:

```
AddHandler imap-file map
```

Finally, this directive supports `.var` files, which are associated with finding the language specified by a web browser client:

```
AddHandler type-map var
```

Part of the process includes output filters. For example, the following `AddOutputFilter` directive looks in web pages with `.shtml` extensions for Server Side Includes.

```
AddOutputFilter INCLUDES .shtml
```

Error Messages

On a web server, if you have an error, you get a message associated with a specific web page. Figure 25.3 illustrates the error message associated with the HTML 404 error code, also known as the "file not found" error.

FIGURE 25.3
An HTML 404 Error

The default error directory is `/var/www/error`; the following `Alias` directive associates the `error` directory with those files:

```
Alias /error/ "/var/www/error/"
```

The following modules provide for content negotiation and SSIs in the web pages in the `/var/www/error/` directory:

```
<IfModule mod_negotiation.c>
<IfModule mod_include.c>
```

The following permissions on the /var/www/error directory set the stage for error messages in English, Spanish, German, and French, in that order. You can read more about the other directives in the "Directory Index" section earlier in this chapter.

```
<Directory "/var/www/error">
    AllowOverride None
    Options IncludesNoExec
    AddOutputFilter Includes html
    AddHandler type-map var
    Order allow,deny
    Allow from all
    LanguagePriority en es de fr
    ForceLanguagePriority Prefer Fallback
</Directory>
```

This works hand in hand with HTML error codes. The page a user sees depends on the error code and the web page defined by the following ErrorDocument directives:

```
ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
```

Browser Customization

When a web browser asks for a web page, it tells Apache what kind of browser it is. The BrowserMatch directive helps you customize the response to different web browsers.

```
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
```

The first two commands create special responses for older browsers; `Mozilla/2` corresponds to Netscape 2.*x*, and `MSIE 4\.0b2` corresponds to Microsoft Internet Explorer 4.*x*. These browsers do not conform to the current HTTP 1.1 standard. The last three commands force HTTP 1.0–level responses to the specified web browsers.

There is a special issue with Microsoft WebFolders, which does not properly handle WebDAV databases. This issue is addressed with the following `BrowserMatch` directives:

```
BrowserMatch "Microsoft Data Access Internet PublishingProvider"
➥ redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully
```

Server Reports

You can send reports on the status and configuration information on your Apache server with various server reports. For example, the following command stanza, when activated, can give you the current status of Apache:

```
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .your-domain.com
#</Location>
```

I would activate it with the following commands; otherwise, the `Deny from all` command would stop all traffic to the `http://servername/server-status` address. In this case, my LAN is on the 192.168.13.0/24 network.

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 192.168.13.0/24
</Location>
```

You can see the result from another computer on my LAN through a different web browser in Figure 25.4.

You can get similar reports on your Apache configuration when you properly activate the following commands:

```
#<Location /server-info>
#    SetHandler server-info
#    Order deny,allow
#    Deny from all
#    Allow from .your-domain.com
#</Location>
```

These commands are direct from the default `httpd.conf` file; remember to set `Allow from` *`your_network_address`*, similar to what I did in the previous stanza. When you do, you can see the results remotely, as shown in Figure 25.5.

FIGURE 25.4
Checking server status remotely

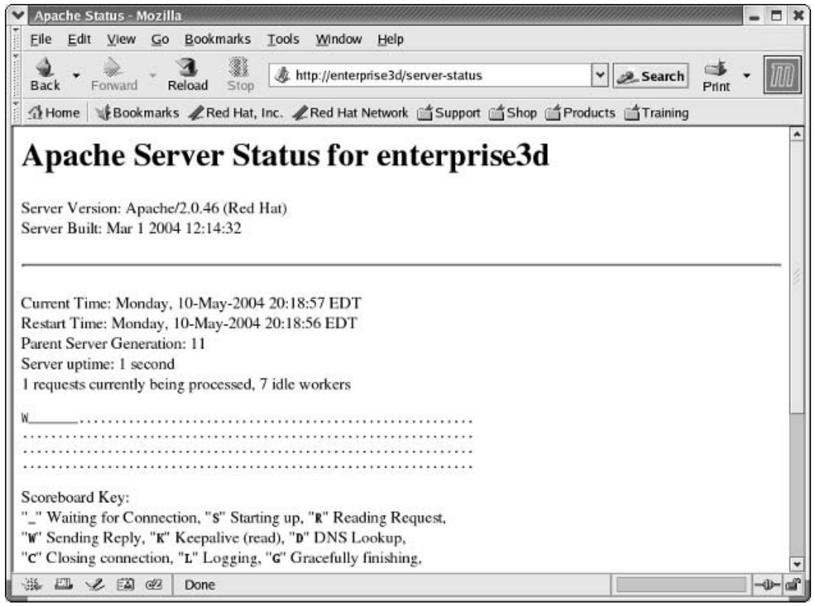
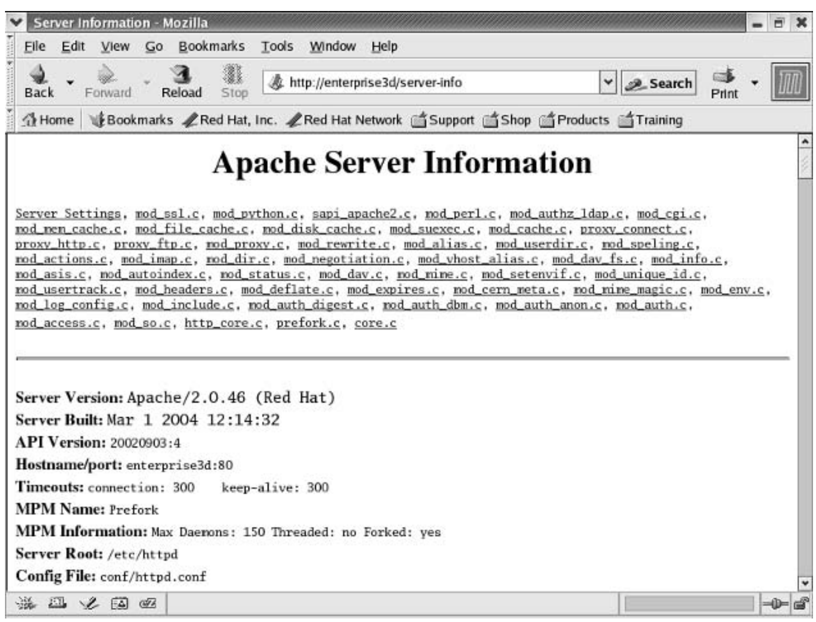


FIGURE 25.5
Checking server configuration remotely



Proxy Server

Apache includes its own proxy server. You can set Apache to cache and serve requested web pages on local networks or all users. The basic commands are shown here; I've changed them a bit to apply the proxy server to my LAN with a network address of 192.168.13.0/24:

```
#<IfModule mod_proxy.c>
#ProxyRequests On
#
#<Proxy *>
#    Order deny,allow
#    Deny from all
#    Allow from 192.168.13.0/24
#</Proxy>
```

If you have multiple proxy servers, you should activate the following `ProxyVia` directive, which supports searches through a chain of proxy servers using HTTP 1.1:

```
#ProxyVia On
```

A proxy server has no purpose unless you configure a cache. Table 25.9 describes the series of special directives associated with caches. If you set up a proxy server, you may want to add more settings; for example, you may want to configure a `CacheSize` to protect your system from becoming overloaded.

```
#<IfModule mod_disk_cache.c>
#   CacheEnable disk /
#   CacheRoot "/var/cache/mod_proxy"
#</IfModule>
```

Table 25.9: Apache Cache Directives

Directive	Description
`CacheDefaultExpire`	Sets the time to cache a document, in seconds.
`CacheEnable`	Supports caching of a specified directory.
`CacheGcInterval`	Configures the time between attempts to clear old data from a cache, in hours.
`CacheLastModifiedFactor`	Sets the expiration time for files in the cache. If there is no expiration date and time associated with a web page, Apache sets it relative to the amount of time since the last known change to that page.
`CacheMaxExpire`	Selects the maximum time in seconds to cache a document.
`CacheRoot`	Configures the default directory with the proxy server cache.
`CacheSize`	Sets the size of the cache, in kilobytes.

Virtual Hosts

One of the strengths of Apache 2.0.*x* is its ability to set up multiple websites on a single IP address. This is possible with the concept of *Virtual Hosts*. Many web hosting companies use Virtual Hosts to serve several websites from a single server.

Older versions of Apache supported only IP-based Virtual Hosts, which required separate IP addresses for each website configured through your Apache server. Apache 2.0.*x* supports name-based Virtual Hosts.

In this scheme, DNS servers map multiple domain names, such as `www.mommabears.com` and `www.sybex.com`, to the same IP address, such as 10.111.123.45. You can set up `httpd.conf` to recognize the different domain names and serve the appropriate website.

NOTE *You can't always use the name-based scheme; it may have problems with older clients, such as Netscape 2.0 and Internet Explorer 4.0 browsers. These browsers cannot handle a lot of information associated with the current HTTP 1.1 standard.*

The following code is an example of how to configure two Virtual Hosts, in this case for `www.sybex.com` and `www.mommabears.com`:

```
NameVirtualHost *:80
```

This `NameVirtualHost` directive listens to requests to all IP addresses on the local computer. It specifies the standard TCP/IP port for web pages, 80. If you want to map several websites to the same IP address, you'll want to substitute it for * in this section. Make sure to use the IP address of your local web server computer.

```
NameVirtualHost 10.111.123.45:80
<VirtualHost 10.111.123.45:80>
    ServerAdmin webmaster@sybex.com
    DocumentRoot /www/site1/sybex.com
    ServerName sybex.com
    ErrorLog logs/sybex.com-error_log
    CustomLog logs/sybex.com-access_log common
</VirtualHost>
```

The directives in the `www.sybex.com <Virtual Host 10.111.123.45:80>` container supersede any settings made earlier in the `httpd.conf` file. You can customize each Virtual Host by adding the directives of your choice.

```
<VirtualHost 10.111.123.45:80>
    ServerAdmin webmaster@mommabears.com
    DocumentRoot /www/site2/mommabears.com
    ServerName mommabears.com
    ErrorLog logs/mommabears.com-error_log
    CustomLog logs/mommabears.com-access_log common
</VirtualHost>
```

As you can see, the settings for the `mommabears.com` website are similar; remember, relative directories depend on the `ServerRoot` directive.

Customizing Apache Modules

There are a number of Apache module-specific configuration files in the `/etc/httpd/conf.d` directory, installed through some of the module RPMs described earlier in the "Packages" section. They are included in the basic Apache configuration courtesy of the `Include conf.d/*.conf` directive in the main `httpd.conf` file.

Most significant in this list is the `ssl.conf` file, as you can configure secure areas for your websites in this file. These module files are summarized in Table 25.10.

Table 25.10: Apache Module Configuration Files in */etc/httpd/conf.d*

File	Description
`authz_ldap.conf`	Supports access to authentication via LDAP; the default version of this file includes the modules and associated LDAP authentication commands.
`perl.conf`	Incorporates a Perl interpreter; supports the use of Perl commands and scripts.
`php.conf`	Incorporates a PHP scripting language interpreter.
`python.conf`	Configures a Python interpreter; allows the use of Python commands and scripts.
`squirrelmail.conf`	Supports the use of the Squirrelmail web-based e-mail reader.
`ssl.conf`	Adds Secure Socket Layer (SSL) support; uses TCP/IP port 443 by default. Includes several directives for certificates and encryption methods.
`webalizer.conf`	Allows access to the Webalizer log file reader.
`welcome.conf`	Defaults to the standard Red Hat Enterprise Linux Test Page shown in Figure 25.1.

One useful tool is the Webalizer. It's included with Apache and can help webmasters monitor and analyze the traffic to their websites. By default, the commands in the `webalizer.conf` file support access from the computer that we've configured as a web server and allow access to website data similar to that shown in Figure 25.6.

Secure Apache Virtual Hosts

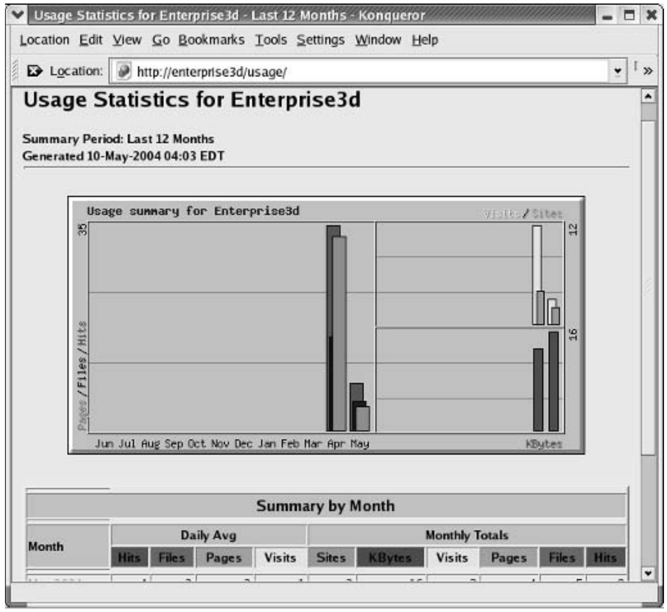
As people need access to public websites, they also need web pages that can be kept secure. This includes the encryption associated with the files in the `/etc/httpd/conf.d` directory. In essence, you configure Virtual Hosts in the `ssl.conf` file in this directory, using the same basic commands as used in the standard `httpd.conf` file.

This file includes a number of secure modules that you can use as configured.Once you've configured a secure virtual host, you can access the secure website through your browser, based on the secure HTTP protocol, also known as HTTPS. You can then access secure websites through your browser, such as `https://www.mommabears.com`.

The first key directive sets the TCP/IP port number for secure services. This is the same `Listen` directive that we described in the `httpd.conf` file.

```
Listen 0.0.0.0:443
```

FIGURE 25.6
Analyzing website traffic data



Now to set up a secure encrypted area of the Sybex website described earlier, we could add the following `VirtualHost` container to the `ssl.conf` file:

```
<VirtualHost 10.111.123.45:443>
    ServerAdmin webmaster@sybex.com
    DocumentRoot /www/securesite1/sybex.com
    ServerName sybex.com:443
    ErrorLog logs/sybex.com-ssl_error_log
    TransferLog logs/sybex.com-ssl_access_log
...
</VirtualHost>
```

There are a number of other commands between the starting `<VirtualHost 10.111.123.45:443>` directive and the `</VirtualHost>` directive at the end of the default `ssl.conf` file. If you want to configure multiple secure Virtual Hosts, you can include the default commands from the file, which we summarize in Table 25.11. More detail is available at `httpd.apache.org/docs-2.0/mod`.

TABLE 25.11: APACHE COMMANDS IN */ETC/HTTPD/CONF.D/SSL.CONF*

COMMAND	DESCRIPTION
`LoadModule ssl_module modules/mod_ssl.so`	Loads the Secure Sockets Layer module for Apache.
`Listen 0.0.0.0:443`	Listens to requests from all IP addresses on the standard HTTPS port, 443.

Continued on next page

TABLE 25.11: APACHE COMMANDS IN */ETC/HTTPD/CONF.D/SSL.CONF (continued)*

COMMAND	DESCRIPTION
`# ErrorLog`	Supports error logs for Dynamic Shared Object modules, which support shared program libraries.
`# CustomLog`	Supports custom access logs for Dynamic Shared Object modules, which support shared program libraries.
`AddType application/x-x509-ca-cert .crt`	Allows you to download secure certificates.
`AddType application/x-pkcs7-crl    .crl`	Lets you use the Netscape certificate management system; applies to Mozilla and related browsers.
`SSLPassPhraseDialog  builtin`	Supports built-in passphrases for encryption.
`SSLSessionCache`	Adds a high-performance hash table for encryption.
`SSLSessionCacheTimeout`	Notes the timeout after which a SSL session expires.
`SSLMutex`	Adds a locking mechanism, so two clients don't use the same web server process.
`SSLRandomSeed`	Points to a random number generator.
`SSLCryptoDevice`	Supports hardware accelerators; reviews a list of available accelerators with the `openssl engine` command.
`TransferLog`	Notes a standard location for the access transfer log file.
`SSLEngine on`	Required to support SSL for Virtual Hosts in this file.
`SSLCipherSuite`	Lists encryption protocols that can be negotiated with this secure web server.
`SSLCertificateFile`	Adds the standard certificate; we'll show you how to create your own shortly.
`SSLCertificateKeyFile`	Adds the standard certificate key; we'll show you how to generate your own key shortly.
`#SSLCertificateChainFile`	Notes an alternate location for your SSL certificates.
`#SSLCACertificatePath`	Lists the default directory path for SSL certificates.
`#SSLCARevocationPath`	If you need to revoke some SSL certificates, you can use a directory such as `/etc/httpd/conf/ssl.crl`.
`#SSLCARevocationFile`	You can specify a file in `SSLCARevocationPath` for revoked certificates.
`#SSLVerifyClient`	Sets a verification level for web server clients.
`#SSLVerifyDepth`	Defines the level of allowed certificate authorities.

Other commands in this file include the following stanza, which configures standard environment variables related to SSL web pages:

```
<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>
<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>
```

The following commands downgrades the HTTP protocol level for Internet Explorer clients:

```
SetEnvIf User-Agent ".*MSIE.*" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0
```

Generating Security Keys

This sidebar gives basic instructions on generating a real set of security keys for Apache. Assuming you have the appropriate RPM packages installed, follow these steps:

1. Delete the basic server keys with the following commands:

```
# rm /etc/httpd/conf/ssl.key/server.key
```

```
# rm /etc/httpd/conf/ssl.crt/server.crt
```

2. Navigate to the `/usr/share/ssl/certs` directory.

```
# cd /usr/share/ssl/certs
```

3. Next, generate a new server key.

```
# make genkey
```

You're prompted twice for a special password known as a *passphrase*. Be careful—this case-sensitive password holds the key to the secure information on your web server.

4. You can now set up a request to a CA with the following command:

```
# make certreq
```

You're prompted for your passphrase and administrative information for your server. Once complete, this command creates the following file, which you can send as part of your request to the CA:

```
/etc/httpd/conf/ssl.csr/server.csr
```

5. The CA should respond to you with a file that you can save as `server.crt` in the `/etc/httpd/conf/ssl.crt` directory.

You can make your own unofficial certificate for test purposes by running the `make testcert` command in step 4.

The next time you start Apache, it prompts you for the passphrase. If you don't get it right, Apache does not start.

Finally, there is one more custom log file, using the fields defined earlier in this chapter.

```
CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
```

If you're actually planning to run a secure web server, you'll need a real set of certificate information from a Certificate Authority (CA) such as VeriSign (`www.verisign.com`) or Thawte (`www.thawte.com`). While we provide general instructions for setting up a secure server in the sidebar "Generating Security Keys," details are extensive and beyond the scope of this book. Refer to `httpd.apache.org`, `www.apache-ssl.org`, and *Linux Apache Web Server Administration,* Second Edition (Sybex, 2002) for more information.

Changes you make here are written to the `ssl.conf` file in the `/etc/httpd/conf.d` directory.

User-Based Security

Secure websites are a good idea. They're not hard to implement. They encrypt communication between browsers and servers, so it's a lot more difficult for crackers to see the information that you want to keep secure. Perhaps this includes credit card or U.S. Social Security numbers.

However, secure websites are slower. They require that you keep secure certificates and encryption keys. In theory, this should not be an issue for you, as secure keys are included automatically with the version of Apache that is included with Red Hat Enterprise Linux.

Nevertheless, some server administrators don't want to bother with HTTPS secure websites. Others charge more for this service, as higher levels of security may increase insurance and or legal costs for the service.

You can offer a different level of security for webmasters who use your Apache service. It's possible to set up user-based security to some or all files on a website, which allows access to those with authorized usernames and passwords.

If you want to add user-based security to a website, you'll need to add some `Auth*` directives to a specific directory and then configure web-based usernames and passwords with the `htpasswd` command.

For example, if you wanted to limit access to files in the `/www/site2/mommabears.com/members` directory, you could add the following commands:

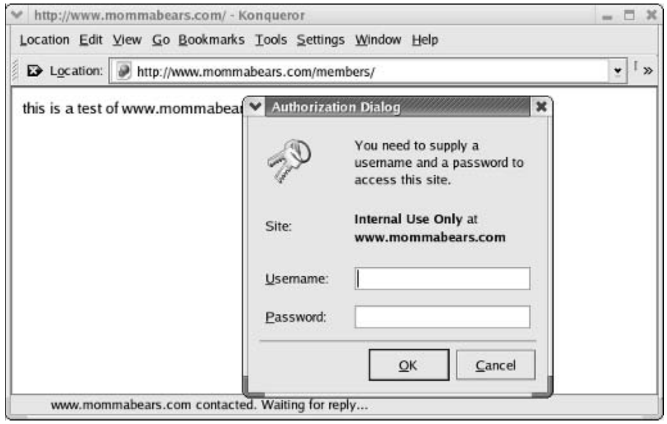
```
<Directory "/www/site2/mommabears.com/members">
     AuthType Basic
     AuthName "Members Only"
     AuthUserFile /etc/httpd/webpass
</Directory>
```

Once you've restarted Apache with the `apachectl restart` command, access is limited to the `/members` subdirectory of the `mommabears.com` website. Thus, when you navigate to `www.mommabears.com/members`, Apache requests that you enter an authorized username and password, as shown in Figure 25.7.

But you haven't configured an authorized username or password yet. Fortunately, it's easy to set up with the `htpasswd` command. Based on the `AuthUserFile` directive shown previously, you can set up the first user (widower) in the file with the following command. The `-c` creates the given file.

```
# htpasswd -c /etc/httpd/webpass widower
```

FIGURE 25.7
Requiring authentication



You're prompted for the password. If you want to add more users to this file, drop the `-c` switch:

```
# htpasswd /etc/httpd/webpass widow
```

Troubleshooting Apache

If you're unable to make a connection to a website configured on a Apache web server, you can check a number of things. Before you begin, check the network. The most common problem on any network is physical; for example, it's good to inspect connectors and cables. Then, check connectivity. You can do so with commands such as `ping`; for more information, see Chapter 16.

Checking Basic Operation

Once you're sure your network is operational, the next step is to see if Apache is running. Start with the following command:

```
# service httpd status
```

You should see a message such as:

```
httpd (pid 3464 3463 3462 3461 3460 3459 3458) is running
```

This tells you that a number of Apache (`httpd`) daemons are running; the number depends on `httpd.conf` directives such as `StartServers`. If you're having a problem, there are three other fairly common messages:

```
httpd is stopped
```

This is fairly simple; try a `service httpd start` command. Rerun the `service httpd status` command. You may also see the following message:

```
httpd is dead but pid file exists
```

In this example, Apache can't start, in part because there is an `httpd.pid` file in the `/var/run` directory. This can happen after a power failure (assuming you don't have an uninterruptible power supply) where Linux never got a chance to erase the `httpd.pid` file. Try deleting the file, and then run the `service httpd start` command. Rerun the `service httpd status` command. You may now see the following message:

```
httpd dead but subsys locked
```

That tells us something else is going wrong. It's time to inspect the log files.

Checking Log Files

The default location for your Apache log files as defined in `httpd.conf` is `/etc/httpd/logs`; however, you'll find this directory linked to a more standard location for log files, `/var/log/httpd`. Remember, you have the freedom to put log files in a different directory by using `CustomLog` directives in a Virtual Host container.

Read the log files in this directory for clues. The variety of errors that you may find is beyond the scope of this book; however, many of the log entries are self-explanatory.

Checking Syntax

The Apache web server includes its own syntax checker. The following command checks the syntax of the main configuration file, `httpd.conf`. If there is a problem, the command

```
# httpd -t
```

often identifies the line number with the problem, such as a misspelled directive. Alternatively, the following command starts Apache in debug mode, which can help you identify additional problems:

```
# httpd -X
```

Checking the Firewall

Sometimes messages just aren't getting through to your web server. That may mean you forgot to let in messages through the standard HTTP port (80) in the firewall. Run an `iptables -L` command to list current firewall rules. Refer to Chapter 17 for more information on this command.

As described with the various firewall utilities (Chapters 3, 4, and 17), you can set up firewalls that automatically allow data through the HTTP port. Remember, if you also serve secure web pages, you should also open the associated port. In this case, for HTTPS, that is port 443. Standard TCP/IP port numbers are defined in `/etc/services`.

Configuring with the Red Hat GUI Apache Tool

Red Hat has developed a GUI tool for configuring Apache, which you can start with the `redhat-config-httpd` command. When you first start the tool in a GUI, you should see the Apache Configuration window, shown in Figure 25.8.

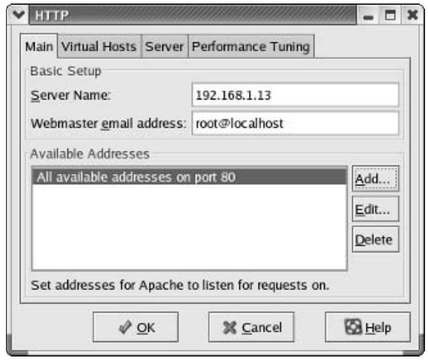


FIGURE 25.8 The graphical Apache configuration utility

As you can see, this utility includes four tabs, which we cover in the following sections. When you finish your changes and click OK, changes are written to your `httpd.conf` file, overwriting any changes you may have made earlier in a text editor.

NOTE *As of this writing,* `redhat-config-httpd` *is still a work in progress. Before I use this utility, I first back up my current* `httpd.conf` *file. After I make changes, I make sure to test the syntax of* `httpd.conf` *with the* `httpd -t` *command. I open* `httpd.conf` *in a text editor to analyze the changes. Nevertheless,* `redhat-config-httpd` *is a great way to learn more about configuring Apache.*

Setting Main Apache Parameters

The basic setup of Apache is straightforward. You're configuring three directives in the Apache configuration window's Main tab.

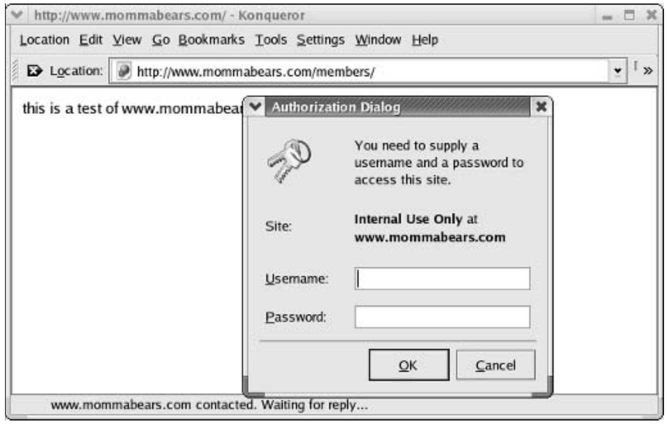
- The Server Name text box corresponds to the `ServerName` directive, which sets the name for the main website for the Apache server. This utility won't work unless you enter the name or IP address of your server in this text box. If you're configuring Virtual Hosts, don't enter any of those domain names in this text box. It is usually best to enter the IP address for your server, to avoid unnecessary traffic to any DNS servers connected to your network.
- The Webmaster Email Address text box corresponds to the `ServerAdmin` directive, which sets the default e-mail address listed by automatically generated web pages. You can see the default setting, `root@localhost`, in Figure 25.8.
- The Available Addresses box sets the TCP/IP ports where Apache listens for requests, using the `Listen` directive. Port 80 is the standard HTTP TCP/IP port, and Apache normally listens to requests from all addresses on the Internet, with the `Allow from all` command.

You can limit the range of computers allowed to view your website. Highlight All Available Addresses On Port 80, and click Edit. This opens the Edit An Address window, shown in Figure 25.9.

For example, Figure 25.9 illustrates limiting access to the network adapter on your computer with an IP address of 192.168.1.14. This changes the `Listen` directive in `httpd.conf` to

```
Listen 192.168.1.14:80
```

FIGURE 25.9 Limiting access to your web server



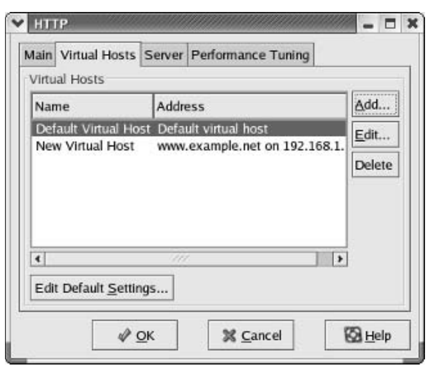
Although you can configure other services using this tool, it may not lead to a fully appropriate result. For example, you can set up a secure web server using the HTTPS protocol. However, the result is saved to the `httpd.conf`, not the `ssl.conf` file described earlier.

For secure pages (HTTPS), click the Add button. This opens the Add New Address window, which looks almost identical to Figure 25.9. You can then enter the IP address of the desired network adapter and the TCP/IP port associated with HTTPS, 443. When you've completed your desired changes, click the Virtual Hosts tab.

Configuring Virtual Hosts

Next, you can start configuring Virtual Hosts within Apache. If you haven't already done so, start the `redhat-config-httpd` utility and click the Virtual Hosts tab. The default view is shown in Figure 25.10.

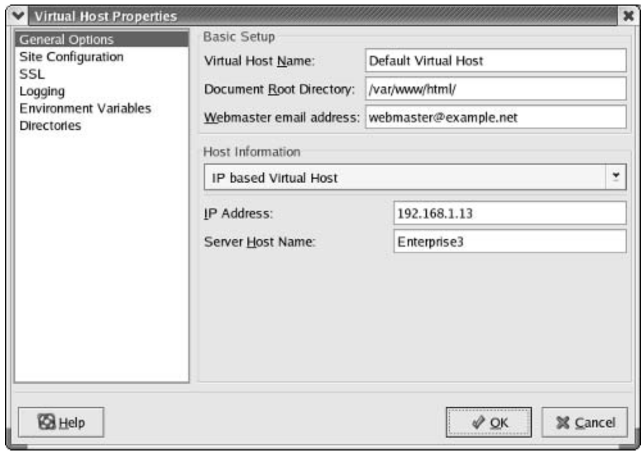
FIGURE 25.10 The Virtual Hosts tab



The Default Virtual Host settings associated with the default `httpd.conf` file, as well as a configuration based on `www.example.net`, are shown. If you want to know more about the default settings, click Edit or Edit Default Settings and analyze the properties window. However, we're focused on

creating a Virtual Host for a real website, so click Add. This opens the Virtual Host Properties window, shown in Figure 25.11.

FIGURE 25.11
Configuring a virtual host



As you can see, there are six sections in this window: General Options, Site Configuration, SSL, Logging, Environment Variables, and Directories.

General Options

Every Virtual Host includes General Options, similar to those shown in Figure 25.11. In that figure, we've filled in some basic parameters for a website named `example.net`.

As described earlier, you can set up multiple Virtual Hosts on a single IP address using the IP-based Virtual Host setting. The alternative, name-based Virtual Hosts, requires an IP address for each website configured through your Apache server.

Site Configuration

Next, select the Site Configuration option on the left side of the window. This opens a list of directory pages and error file settings, as shown in Figure 25.12.

When users look for your website, they're taken to the directory associated with the `DocumentRoot` directive. As you can tell in Figure 25.11, that's the `/var/www/html` directory. It looks for one of the filenames shown in the Directory Page Search List box: `index.php`, `index.html`, `index.htm`, or `index.shtml`.

The Error Pages shown at the bottom of the window display Apache's response to various HTTP errors. For example, the highlighted error, *file not found*, is associated with HTTP error code 404. The default behavior refers to `ErrorDocument` directives in `httpd.conf`. If you want special error pages, you can create special `ErrorDocument` directives for this particular Virtual Host. To do so, highlight the error code of your choice and click Edit. This opens the redhat-config-httpd window, shown in Figure 25.13.

FIGURE 25.12
Site configuration settings

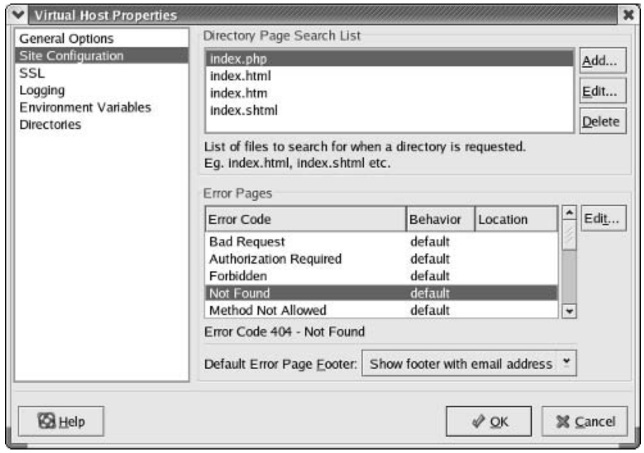
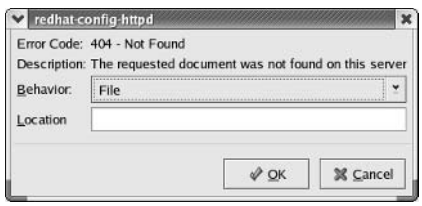


FIGURE 25.13
Changing error code behavior



You can point the user in three directions in the Behavior line for Error Code 404: Default points to the standard `ErrorDocument` directive in `httpd.conf`; File allows you to specify the web page of your choice; and URL lets you set the location of the desired error message online.

Finally, the Default Error Page Footer shown in Figure 25.12 specifies the information associated with each error page. The standard footer is based on the `bottom.html` file in the `/var/www/error/include` directory. You can choose to not show the footer at all, or you can show it with or without an e-mail address.

SSL

Next, select the SSL option on the left side of the window. This opens a series of options associated with the Secure Socket Layer, as shown in Figure 25.14. When you install the Apache `mod_ssl-*` RPM, you get a series of fake keys in the `/etc/httpd/conf` directory, which are shown in the figure.

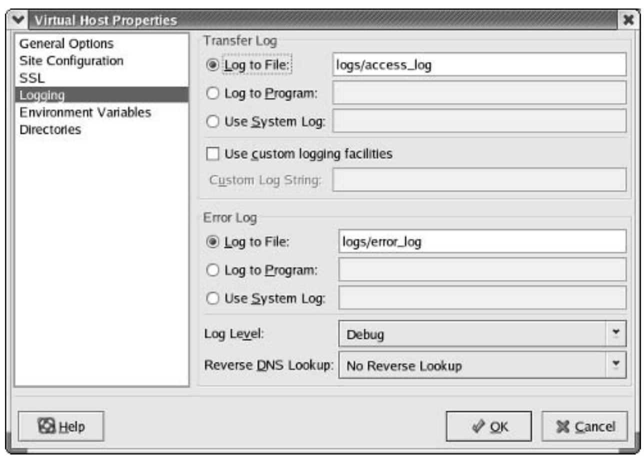
You'll need to create a real set of certificate data, as described earlier in the "Generating Security Keys" sidebar.

LOGGING

Next, select the Logging option on the left side of the window. This opens a series of options associated with logging and log files, as shown in Figure 25.15.

FIGURE 25.14
Secure Socket Layer settings

FIGURE 25.15
Virtual Host logging



The default log files are shown in the figure; the path is relative to the `ServerRoot` directive, normally `/etc/httpd`. Naturally, you may want to specify log files in special directories associated with the Virtual Host, such as `www.example.net/logs/access_log`.

You can specify the information that goes into this log file in the Custom Log String text box. The information here is associated with the `LogFormat` directive described earlier in this chapter.

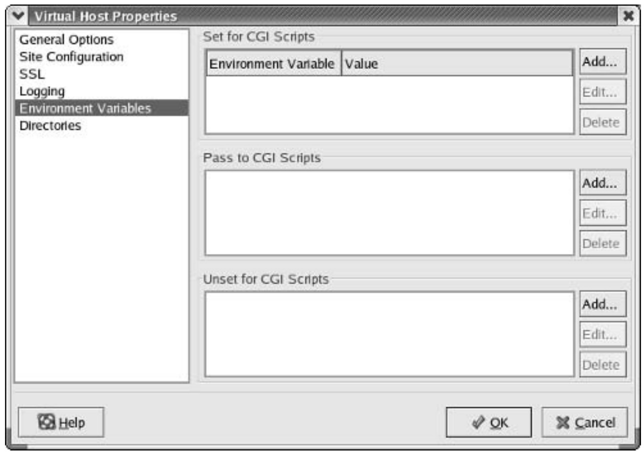
The options available in the Log Level drop-down list match those described earlier for the `LogLevel` directive: Emergency, Alert, Critical, Error, Warn, Notice, Info, and Debug.

You may want to make sure the Reverse DNS Lookup setting is set to No Reverse Lookup. Unless you have a reliable and speedy connection to a DNS server, finding the fully qualified domain names associated with an IP address could hurt your web server's performance.

Environment Variables

Next, select the Environment Variables option on the left side of the window. This opens a group of settings where you can set environment variables associated with CGI or SSI scripts, as shown in Figure 25.16.

Figure 25.16
Environment variables settings

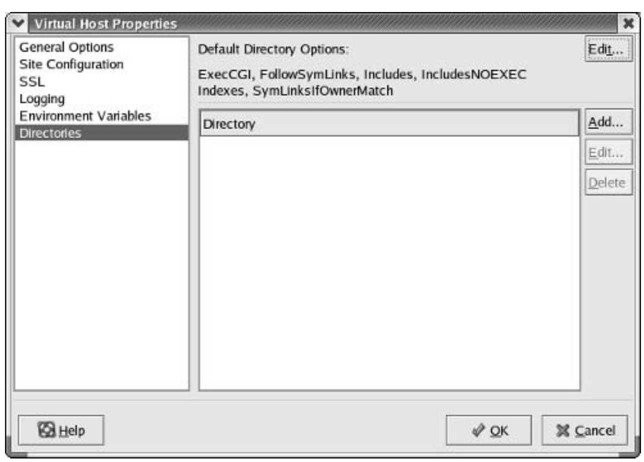


While the principle is the same as regular environment variables in the shell, what you set here applies only to CGI and or SSI scripts.

Directory Options

Finally, select the Directories option on the left side of the window. This opens a group of settings where you can set the `Options` directive for various directories, as shown in Figure 25.17.

Figure 25.17
Directories options



The Options for the default directory are shown in Figure 25.17: ExecCGI, FollowSymLinks, Includes, IncludesNOEXEC, Indexes, and SymLinuxIfOwnerMatch (they are explained back in Table 25.5). You can edit the default settings by clicking the Edit button in the upper-right corner of the window.

You can specify Options for other directories. Click Add to open the Directory Options window shown in Figure 25.18. The options in this window are explained in Table 25.12.

FIGURE 25.18
Setting Options on a new directory

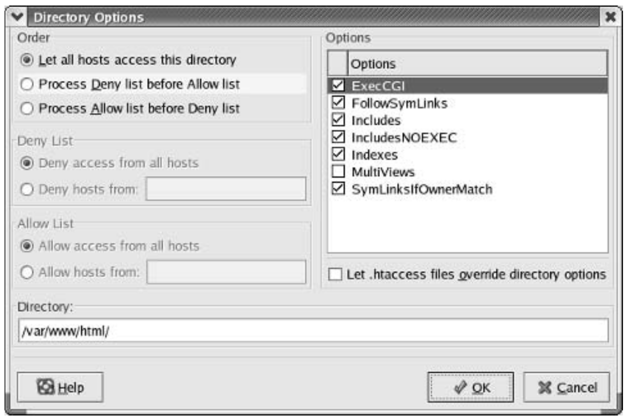


TABLE 25.12: SELECTIONS IN THE DIRECTORY OPTIONS WINDOW

SELECTION	DESCRIPTION
Order	Sets the order of directives; the options are `Allow from all`; `Order deny,allow`; and `Order allow,deny`.
Deny List	If you're not allowing in all hosts, you can deny access to this directory to some or all hosts, by domain name or IP address.
Allow List	If you're not allowing in all hosts, you can allow access to this directory to some or all hosts, by domain name or IP address.
Directory	Specifies the directory to which the `Options` directive is to be applied.
Options	The settings associated with the `Options` directive.
`.htaccess`	If you activate this setting, the `AllowOverride` directive is added to this directory.

Configuring the Server

There are some basic settings associated with each Apache server. Return to the Apache Configuration window and click the Server tab. The information should look similar to Figure 25.19. These settings are summarized in Table 25.13.

FIGURE 25.19
Apache configuration server settings

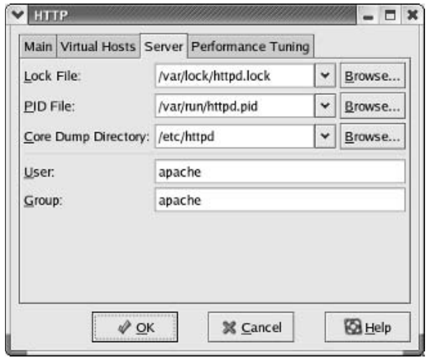


TABLE 25.13: APACHE CONFIGURATION SERVER SETTINGS

SETTING	DESCRIPTION
Lock File	The file is opened by Apache when it starts.
PID File	Another file opened by the Apache when it starts. Includes the PIDs associated with open `httpd` daemons.
Core Dump Directory	Specifies the directory for core dumps, which are used for debugging. Must be writeable by the user associated with the Apache server, normally apache.
User	The username associated with the Apache server.
Group	The group name associated with the Apache server.

Performance Tuning

Several basic performance settings are associated with each Apache server. In the Apache Configuration window, click the Performance Tuning tab. The information should look similar to Figure 25.20. These settings are summarized in Table 25.14.

FIGURE 25.20
The Performance Tuning tab

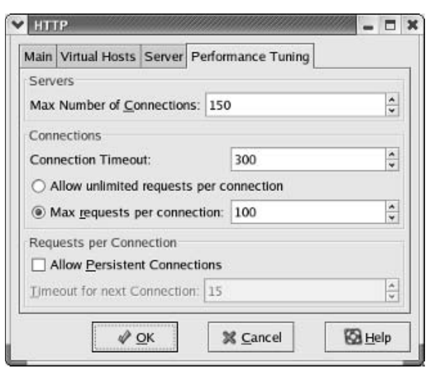


TABLE 25.14: APACHE CONFIGURATION PERFORMANCE SETTINGS

SETTING	DESCRIPTION
Max Number Of Connections	Corresponds to the maximum number of clients who can connect to your web server simultaneously; sets the `MaxClients` directive.
Connection Timeout	Sets the time the web server waits for further communication from a client browser, in seconds; sets the `TimeOut` directive.
Requests Per Connection	Limits the number of requested items per connected browser; sets the `MaxRequestsPerChild` directive.
Allow Persistent Connections	Keeps connections open to a browser, independent of `TimeOut`; if selected, the `KeepAlive` directive is set to true.
Timeout For Next Connection	Sets the time which Apache waits for the next request from a client, if `KeepAlive` is true; sets the `KeepAliveTimeout` directive.

Incorporating the Red Hat Content Accelerator

The Red Hat Content Accelerator is an alternative web server. Also known as TUX, this web server is designed to manage static web content quickly, because its settings reside directly in the Linux kernel. While this tool can also manage dynamic web pages, Red Hat recommends using the Content Accelerator for static pages in concert with Apache for dynamic pages.

This tool is still a work in progress, since TUX can work with Apache only if they're both loaded on the same computer. That's a less than convenient situation for larger websites, which often require several servers, often in different geographic locations. Since the package name for the Red Hat Content Accelerator is still TUX, we'll use the terms interchangeably in this section.

NOTE *TUX stands for a Threaded Linux web server. Incidentally, it is also the name for the Linux mascot penguin.*

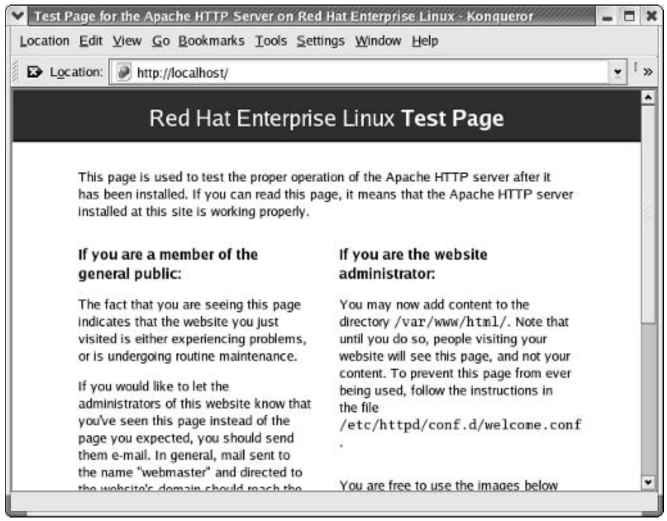
Installing and Starting TUX

Normally, the Red Hat Content Accelerator (TUX) and Apache can't run simultaneously. It's installed automatically when you install the Web Server package group. Just in case, you can check for and install the `tux-*` package using the appropriate `rpm` commands. The default TUX configuration file is `/etc/sysconfig/tux`. To see if TUX works on your computer, we'll configure and start it after stopping Apache, using the following steps:

1. Set up an `index.html` file in the default `DOCROOT` directory, `/var/www/html`. For the purpose of this exercise, I've modified and copied `/var/www/error/noindex.html`, which you should recognize from Figure 25.1.
2. Stop the Apache server with the `apachectl stop` command.
3. Start the TUX server with the `service tux start` command.
4. Navigate to the localhost in the browser of your choice.

You can review the result in Figure 25.21. Since we've stopped the Apache daemon, we know it's TUX at work serving the noted web page.

FIGURE 25.21
TUX at work



In the figure, we see TUX serving a standard `index.html` page from `/var/www/html`. So, TUX works in Linux. We'll take a brief look at the mechanics in the following sections.

NOTE *Red Hat recommends that you configure the* `DOCROOT` *associated with TUX in a separate RAID partition. This corresponds to the Apache* `DocumentRoot` *directive. For more information on RAID, see Chapter 14.*

Deciphering the Content Accelerator Configuration

Since the Content Accelerator configuration is part of the kernel, the default settings aren't in a standard configuration file; as described earlier, they're located in `/etc/sysconfig/tux`. This file includes several parameters, which are described in Table 25.15.

TABLE 25.15: TUX CONFIGURATION PARAMETERS

PARAMETER	DESCRIPTION
`TUXTHREADS`	Defines the allowed number of kernel threads; do *not* set this higher than the number of CPUs on your computer. (Hyperthreaded CPUs reportedly do not work well as a multiprocessor.)
`DOCROOT`	Sets the top-level directory for requests to the web server; corresponds to the Apache `DocumentRoot` directive. Browsers expect to find a home page file such as `index.html` in this directory. The standard location is `/var/www/html`.
`LOGFILE`	Assigns a location for the TUX log file, normally `/var/log/tux`.

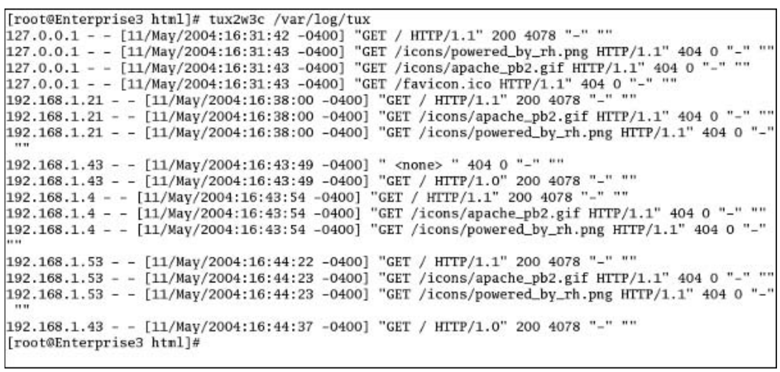
Continued on next page

Table 25.15: TUX Configuration Parameters *(continued)*

Parameter	Description
`DAEMON_UID`	Associates a user ID with TUX. The default is nobody.
`DAEMON_GID`	Associates a group ID with TUX. The default is nobody.
`CGIROOT`	Defines the directory for CGI scripts, if required (Tux is advertised as being faster for *static* data, after all).
`MAX_KEEPALIVE_TIMEOUT`	Sets a timeout value for connections, in case of network problems.
`TUXMODULES`	Defines modules for dynamic TUX data.
`MODULEPATH`	Sets the directory with Content Accelerator application program interface modules.

TUX also includes a series of log files, located in a file named `/var/log/tux`. However, these log files are compressed in binary format. To read them, you need the `tux2w3c` command. One result is shown in Figure 25.20, which illustrates several connections within my network: from the local computer (127.0.0.1) and from four computers from within my private LAN (192.168.1.0).

Figure 25.22
An interpreted TUX log file



```
[root@Enterprise3 html]# tux2w3c /var/log/tux
127.0.0.1 - - [11/May/2004:16:31:42 -0400] "GET / HTTP/1.1" 200 4078 "-" ""
127.0.0.1 - - [11/May/2004:16:31:43 -0400] "GET /icons/powered_by_rh.png HTTP/1.1" 404 0 "-" ""
127.0.0.1 - - [11/May/2004:16:31:43 -0400] "GET /icons/apache_pb2.gif HTTP/1.1" 404 0 "-" ""
127.0.0.1 - - [11/May/2004:16:31:43 -0400] "GET /favicon.ico HTTP/1.1" 404 0 "-" ""
192.168.1.21 - - [11/May/2004:16:38:00 -0400] "GET / HTTP/1.1" 200 4078 "-" ""
192.168.1.21 - - [11/May/2004:16:38:00 -0400] "GET /icons/apache_pb2.gif HTTP/1.1" 404 0 "-" ""
192.168.1.21 - - [11/May/2004:16:38:00 -0400] "GET /icons/powered_by_rh.png HTTP/1.1" 404 0 "-"
 ""

192.168.1.43 - - [11/May/2004:16:43:49 -0400] " <none> " 404 0 "-" ""
192.168.1.43 - - [11/May/2004:16:43:49 -0400] "GET / HTTP/1.0" 200 4078 "-" ""
192.168.1.4 - - [11/May/2004:16:43:54 -0400] "GET / HTTP/1.1" 200 4078 "-" ""
192.168.1.4 - - [11/May/2004:16:43:54 -0400] "GET /icons/apache_pb2.gif HTTP/1.1" 404 0 "-" ""
192.168.1.4 - - [11/May/2004:16:43:54 -0400] "GET /icons/powered_by_rh.png HTTP/1.1" 404 0 "-"
""

192.168.1.53 - - [11/May/2004:16:44:22 -0400] "GET / HTTP/1.1" 200 4078 "-" ""
192.168.1.53 - - [11/May/2004:16:44:23 -0400] "GET /icons/apache_pb2.gif HTTP/1.1" 404 0 "-" ""
192.168.1.53 - - [11/May/2004:16:44:23 -0400] "GET /icons/powered_by_rh.png HTTP/1.1" 404 0 "-"
 ""

192.168.1.43 - - [11/May/2004:16:44:37 -0400] "GET / HTTP/1.0" 200 4078 "-" ""
[root@Enterprise3 html]#
```

Combining TUX and Apache

You can set both TUX and Apache to run simultaneously, as long as they're listening on different TCP/IP ports. The changes you need to make to the Apache `httpd.conf` file are simple; they involve two directives.

The `Listen` directive tells Apache about the computers and ports to check for input. Normally, it's set to listen to the standard HTTP port, with a `Listen 80` command. If you're using TUX on the same computer, make it listen locally with this command:

```
Listen 127.0.0.1:8080
```

This corresponds to the way TUX looks for a port number in the `clientport` file in the TUX kernel settings directory: `/proc/sys/net/tux`. This file should have one line, which points to port 8080.

Now, assuming you're using Apache Virtual Hosts, you'll want to specify the IP address associated with your web server through the `NameVirtualHost` directive. You may have already done so earlier in this chapter. Substitute the IP address for your Web server computer.

```
NameVirtualHost 192.168.13.64:80
```

This corresponds to the standard TUX server port, which is located in the `/proc/net/tux/0/listen/0` file. This value in this file should already be set to

```
http://0.0.0.0:80
```

which listens to requests from all IP addresses on TCP/IP port 80.

Once you've made this small changes, you're ready to set TUX and Apache to work together; if you had stopped Apache per the earlier instructions, you should now be able to start it with the following command:

```
# apachectl start
```

Introducing Caching Services

Enterprises often have to work with large numbers of users who connect to the Internet. The more who connect, the more your business pays for Internet access. One thing that can save money is caching services. If you cache frequently used content, then different people who access the same Linux HOWTOs from within your network can get these documents locally. This also makes it faster for users to get to those documents; and that can make you look like a hero.

While you could use the Apache proxy to cache previously used content, the leader in caching services on Linux is the Squid Proxy service. The content you collect is stored in an area known as a *Harvest Cache*. While you can get more information on this service from `www.squid-cache.org`, it's fairly easy to configure Squid.

Squid Hardware

Generally, when you're configuring a computer for a Harvest Cache, the focus is on the storage media. You'll want the largest and fastest hard drives available, with faster seek times. Since all a Squid computer is doing is looking for and serving files, the speed of the CPU is less important.

If you have a larger network that depends on caching, you'll want some form of redundancy, such as in a form of a backup. Squid allows you to configure multiple servers in a parent/child/sibling relationship, where one server can take over for another. Otherwise, if your system fails, your users will experience a drop in effective access speed to the Internet.

All your users' requests for Internet content go through the computer or router that serves as your gateway. Naturally, it's most efficient to locate a Squid computer on or near this computer.

Squid Configuration

The `squid` RPM is installed by default with the Web Server package group. The main Squid configuration file, `/etc/squid/squid.conf`, can be intimidating. It's over 3000 lines long! Fortunately, it's easy to configure. All you need to do is add three lines.

1. First, add a command that names your computer as the host of the proxy server.

   ```
   visible_hostname Enterprise3
   ```

2. Add an `http_access` directive to name the local area network; one example is as follows:

   ```
   http_access allow lan_network
   ```

3. Add your LAN to the Squid access control list. Now that you've set the `http_access` directive to allow access by your `lan_network`, you can use that variable here:

   ```
   acl lan_network src 192.168.1.0/255.255.255.0
   ```

The default version of the `squid.conf` file includes several other examples of each directive, which can help you understand how they work in detail.

TIP *If there are limits on your hard drive space, and you want to make sure that the Squid cache doesn't overwhelm the rest of this computer, you should configure the* `/var/spool/squid` *directory on an independent partition.*

Activation

Once you've configured `squid.conf`, you can activate the Squid service. First, you'll need to create the cache directories with the following command:

```
# squid -z
```

This configures subdirectories in `/var/spool/squid` for storing the Harvest Cache. The other commands for starting Squid should be familiar to you, especially if you've worked with other services in this book. These commands start the Squid service and then ensure that it starts the next time you boot Linux in runlevels 3 and 5.

```
# service squid start
# chkconfig --level 35 squid on
```

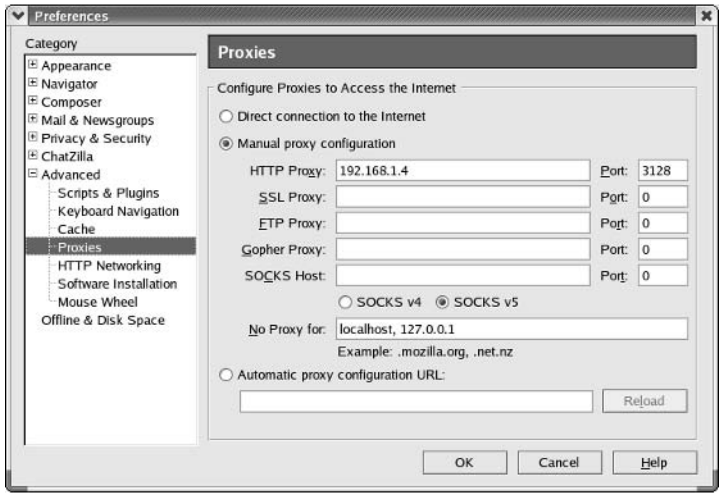
TIP *Some Squid Proxy servers are located on the same computer as your network's gateway to the Internet. As such, it probably has two or more network cards. Generally, a firewall is disabled on the network card that is directly connected to your LAN. If you have to configure a firewall on this network card, make sure to allow traffic through TCP/IP port 3128.*

Configuring Clients on Squid

Now configuring clients to use the Squid service is fairly elementary. Client web browsers allow you to connect to a specified Squid Proxy computer. As this book is focused on servers, and there are sev-

eral different browsers included with Red Hat Enterprise Linux 3, we don't explore this option in detail. One example from the Mozilla web browser is shown in Figure 25.23.

FIGURE 25.23
Configuring Mozilla for Proxy service



Summary

Linux is built for networking. In this age of the Internet, that means that Linux is built as an operating system that works with web servers. You can set up a number of different web servers on Linux, including Apache, TUX, AOLServer, BOA, Zeus, and more.

Apache is the most popular web server on the Internet. With Apache version 2.0.*x*, you can now set up multiple Virtual Hosts on the same web server, using a single IP address.

The main Apache configuration file, `httpd.conf`, is long but not that complex. We've analyzed it in three different sections: the global environment, which governs the operation of the server as a whole; parameters for the main server, which serve as defaults, and Virtual Hosts, where you can configure as many websites as your hardware can handle.

Red Hat recommends an alternative to Apache for static web pages, known as the Red Hat Content Accelerator. Formerly known as TUX, this web service is in many ways faster, because it resides directly in the Linux kernel. It's fairly easy to make these two web services work together. Red Hat recommends using TUX as the primary web server, referring to Apache for dynamic content.

One RPM included with the Web Server package group can help an enterprise save money on a shared Internet connection. The Squid Proxy service caches frequently used content so common requests don't always have to go to the Internet. It's easy to configure; all you need to do is add three commands to the `squid.conf` file.

In the next chapter, we'll show you how you can set up the open-source MySQL packages for working with databases.

Chapter 26

Setting Up MySQL for Databases

COMPUTERS ARE USED TO PROCESS large amounts of data. This data—whether it be engineering studies, economic forecasts, or time card information—often resides in databases.

There are a number of database packages available for Linux. Most use the Structured Query Language (SQL) for creating and managing databases. Many database systems incorporate SQL into their names, such as the two included with Red Hat Enterprise Linux 3: MySQL and PostgreSQL. Other third parties, such as Sybase and Oracle, support Linux with their own database systems. Even Microsoft has its own SQL server. The version of MySQL included with Red Hat Enterprise Linux 3 is released with an open-source license and is therefore a preferred option of many in the Linux community.

This chapter is just a brief overview of MySQL; we can't go into the same detail of the 1,200-page MySQL reference manual in this chapter (you can download this excellent manual from `www.mysql.com`). However, we do show you how to install the MySQL database software, configure MySQL for basic operation, and set up and manage a basic MySQL database. This chapter covers the following topics:

- Installing the MySQL packages
- Analyzing the MySQL configuration files
- Managing a MySQL database

Installing the MySQL Packages

There are two package groups related to SQL: SQL Database and MySQL Database. You can install either to activate the SQL system associated with each database. Alternatively, you can install a third-party package. For example, the Oracle Database 10*g* (Oracle 10*g*) system is optimized for Red Hat Enterprise Linux 3, and can be downloaded from `www.oracle.com`.

NOTE *There is some controversy associated with the use of MySQL version 3.23.58 instead of 5.x. While MySQL version 5.x is released under the GPL, there are some parts of the MySQL client libraries are released under other licenses.*

The SQL and MySQL Package Groups

If you haven't already done so during the installation process, you can install either of the SQL-related package groups using the Package Management tool. While it's a necessary part of the process, it is not enough. As of this writing, the `mysql-server` RPM is not included with the standard Red Hat Enterprise Linux 3 CDs.

Just to review, if you've organized the Red Hat Enterprise Linux 3 installation files on the `/mnt/inst` directory (this can be mounted over a network connection), you can use this source with the following command:

```
# redhat-config-packages --tree=/mnt/inst
```

You can then install one or both of these package groups, as shown in Figure 26.1. Click Details next to either group to review and select individual packages. We review these packages in Table 26.1.

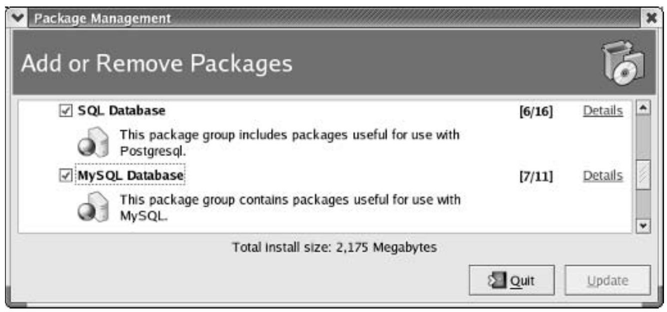


FIGURE 26.1 Installing SQL package groups

TABLE 26.1: SQL DATABASE PACKAGE GROUPS

PACKAGE	DESCRIPTION
`perl-DBD-Pg*`	Adds a version of the Perl Database Interface module for PostgreSQL
`perl-DB_File*`	Supports database file modules for Perl
`postgresql-odbc`	Includes the drivers to access a PostgreSQL server using ODBC
`rh-postgresql`	Installs the basic PostgreSQL client programs and libraries
`rh-postgresql-contrib`	Adds packages contributed to help manage PostgreSQL databases
`rh-postgresql-docs`	Includes extensive documentation for PostgreSQL
`rh-postgresql-jdbc`	Adds the `.jar` file that supports a Java interface to PostgreSQL
`rh-postgresql-pl`	Supports connections to Perl, Tcl, and Python interfaces
`rh-postgresql-python*`	Adds a module for Python code
`rh-postgresql-server`	Installs the basic PostgreSQL server package

Continued on next page

Table 26.1: SQL Database Package Groups *(continued)*

Package	Description
`rh-postgresql-tcl`	Supports connections to the Tcl client library and several shells
`rh-postgresql-test`	Adds test programs
`rhdb-utils*`	Includes miscellaneous tools
`tora*`	Supports the toolkit for Oracle databases
`unixODBC*`	Adds the Open Database Connectivity (ODBC) standard for Linux and Unix computers, which supports communication with SQL databases
`unixODBC-kde`	Supports KDE connections to the ODBC interface to SQL databases

** These are default packages.*

There is also the open source system for Linux databases, built on the MySQL package group. We summarize the packages associated with this group in Table 26.2. You'll notice some overlap in these groups, which reflects the common interfaces to SQL databases.

Table 26.2: MySQL Database Package Groups

Package	Description
`libdbi-db-mysql`	Supports connectivity via the Database Interface library
`mod_auth_mysql*`	Limits access to documents read by a web server
`MyODBC`	Adds an ODBC driver for MySQL databases and works with the `unixODBC` package
`MySQL-python`	Installs a Python interface module to MySQL
`mysql`	Installs the basic MySQL database server
`mysql-bench*`	Adds benchmarking scripts for a MySQL database server
`mysql-devel`	Includes libraries and header files for MySQL applications
`perl-DBD-MySQL`	Installs a MySQL interface for the Perl language
`php-mysql*`	Adds MySQL support to PHP
`qt-mysql*`	Installs a MySQL driver for the Qt toolkit associated with KDE
`unixODBC`	Adds the Open Database Connectivity (ODBC) standard for Linux and Unix computers, which supports communication with SQL databases.

** These are optional packages.*

For the purpose of this chapter, we assume that you want to get the MySQL server up and running. The `mysql-server` package is not included in the standard CDs. Once you've installed the

SQL software with the Package Management tool, you'll need to install the MySQL server using one of the following three sources:

- Using the Red Hat Enterprise Linux 3 server Extras CD.
- Downloading it from the Red Hat Network. This requires an official subscription to Red Hat Enterprise Linux.
- Using a download from a third-party "rebuild" of Red Hat Enterprise Linux 3.

Other SQL Servers

There are a number of other database server systems available for Linux. Many use the SQL language to communicate with databases. The Oracle and Sybase database systems are "certified," for Red Hat Enterprise Linux, as "Premier Partners." We've selected the others noted in this section arbitrarily. I've summarized the options here:

Oracle As it has optimized its database software for Red Hat Enterprise Linux, Oracle's 9*i* and 10*g* may be the most prominent of the options. If you have a standard or premium support contract for Red Hat Enterprise Linux, Oracle (as of this writing) states that it will support your installation of their software, as described in `otn.oracle.com/tech/linux/htdocs/oracleonlinux_faq.html`. You can download and test the Oracle packages of your choice from `www.oracle.com`. Other Red Hat–certified solutions based on Oracle database systems are available from Danlaw (`www.danlaw.com`) and CoSORT (`www.cosort.com`).

Sybase Sybase has also certified its data management solutions, including Adaptive Server Enterprise 12.5 and SQL Anywhere Studio, on Red Hat Enterprise Linux 3. You can download and try Sybase products. However, the associated licenses are not open source, and support requires separate contracts. For more information, see `www.sybase.com`.

Computer Associates Computer Associates includes SQL support for its Advantage Ingres Relational Database server. You can download evaluation copies optimized for Linux from `www3.ca.com/Solutions/Product.asp?ID=1013`.

Hyperion Hyperion includes a number of database options in its Essbase product line. As of this writing, downloads for its Linux-based products are not readily available.

ABAS One European-based database solution, certified for Red Hat Enterprise Linux, is available from ABAS software, which is focused on medium-sized businesses.

Microsoft SQL is enough of a universal standard that Microsoft has had a SQL server for years. While it's more of a GUI product, it still requires extensive text scripting. While it is not supported for Linux, it does support SQL connectivity using ODBC.

There are a number of other database servers available for Linux; however, some are not active. This section represents the current information at the time of this writing. We apologize if your favorite database software has changed since publication or has not been covered here.

Analyzing the MySQL Configuration Files

The latest available production release of MySQL is actually beyond what is included with Red Hat Enterprise Linux. However, stability and open-source licensing are important factors, so using MySQL version 3.23 makes sense on an enterprise-level operating system.

There are a number of sample configuration files available that are included with the `mysql-server` RPM. There is also a default configuration file included with the `mysql` RPM.

Configuring the MySQL server is a rich and complex topic. We can only scratch the surface of the options in this chapter. There are five standard MySQL configuration files that you could use. The last four are in the `/usr/share/doc/mysql-server-*/` directory.

- `/etc/my.cnf` is the default MySQL configuration file. You should make any configuration changes to this file. It's designed for learning purposes.
- `my-small.cnf` is designed for small databases on computers that are also used for a number of other services. You should not use this model for databases with more than a few entries that are used frequently.
- `my-medium.cnf` is designed for moderately sized databases. If you're using Red Hat Enterprise Linux in the enterprise, you probably have significantly more than the minimum RAM required for this operating system (256MB). If you have this kind of memory available, you could conceivably run other services on the same computer.
- `my-large.cnf` is intended for computers that are dedicated to a SQL database. As it assumes up to 512MB of memory for the database, you'll want at least 1GB of RAM on this type of system so it can handle both the operating system and the database application.
- `my-huge.cnf` is intended for databases in the enterprise. Such databases require dedicated servers and 1GB or more of RAM.

These options are highly dependent on the amount of memory, on the speed of your computer, the details and size of your database, and the number of users accessing it on your computer, and the number of users who load and access data from your databases. As your databases and users grow, the performance of your databases may change.

We'll examine each of these configuration files. If you choose to use one of the sample `my-*.cnf` files, you'll first need to copy this file to `/etc/my.cnf`.

For these reasons, you should watch the performance of your database systems carefully. If you find problems, you may want to add more RAM or move your database to a system with additional resources such as multiple CPUs.

NOTE *Databases can become quite large. It can make sense to set up a SQL database directory on a dedicated partition. While a growing database may fill that partition, at least it would then not crowd out the space needed by Red Hat Enterprise Linux 3 to start and run.*

/etc/my.cnf

The default `/etc/my.cnf` file is straightforward. It includes six commands organized in three stanzas. They are similar to stanzas in a Samba configuration file, with functional group names and associated

commands. In this section, we'll analyze the default version of this file, line by line. If you make any changes, you'll want to make sure that the commands in the MySQL start script (`/etc/rc.d/init.d/mysqld`) are consistent.

```
[mysqld]
```

You'll see commands related to the MySQL daemon under this group.

```
datadir=/var/lib/mysql
```

The MySQL server stores data in the directory defined by the `datadir` variable.

```
socket=/var/lib/mysql/mysql.sock
```

The MySQL socket connects the database program, locally or over a network, to MySQL clients.

Note *MySQL is configured to use the InnoDB storage engine. If you don't have an InnoDB database in your system, you'll need to add the* `skip-innodb` *statement to the* `[mysqld]` *stanza.*

```
[mysql.server]
```

You'll see commands related to the MySQL server daemon under this group. The older version of this group was named `[mysql_server]`. If you use MySQL version 4.*x* or above, you'll have to convert this group title to `[mysql.server]`. When you start the MySQL service, it uses the options in this stanza.

```
user=mysql
```

The standard username associated with the MySQL service is mysql. It should be a part of `/etc/passwd`; if you don't find it there, you may not have installed the Red Hat Enterprise Linux `mysql-server` RPM.

```
basedir=/var/lib
```

This represents the top-level directory of the MySQL database. It acts as a root directory on your MySQL system; other directories are relative to this one in this database.

```
[safe_mysqld]
```

This includes the directives cited by the MySQL start script. If you use MySQL version 4.*x* or above, you'll have to convert this group to `[mysql_safe]`.

```
err-log=/var/log/mysqld.log
```

This is the file where MySQL related errors are sent. If you use MySQL version 4.*x* or above, you'll have to replace this with the `log-error` directive.

```
pid-file=/var/run/mysqld/msqld.pid
```

Finally, the `pid-file` defines the process identifier (PID) of the MySQL server while in operation. If the MySQL server is not running, this file should not exist.

NOTE *You can configure user-specific MySQL configuration files; all you need to do is add the configuration commands and directives of your choice to the hidden* `.my.cnf` *file in a specific user's home directory.*

my-small.cnf

In this section, we'll analyze all the commands in the `my-small.cnf` sample MySQL configuration file. When we review other sample MySQL configuration files, we'll refer to this section for the meaning of various commands and directives. Analyzing the active commands and directives in this file, we start with the following group:

```
[client]
```

This group passes directives to clients associated with your MySQL server.

```
port=3306
```

The standard TCP/IP port associated with MySQL is 3306. If you want to change this port number (which could promote security), you have to be sure to change this number in all applicable configuration files for your MySQL clients and servers.

```
socket=/var/lib/mysql/mysql.sock
```

This is the standard socket file that governs communications between MySQL clients and servers, just as defined in the default `/etc/my.cnf` file.

```
[mysqld]
```

When you start the MySQL server, it's governed by the commands defined in the `[mysqld]` stanza.

```
port=3306
socket=/var/lib/mysql/mysql.sock
```

Naturally, clients and servers associated with a MySQL database need to use the same TCP/IP port and socket.

```
skip-locking
```

Multiple clients may access the same database, so this prevents external clients from locking your MySQL server. The `skip-locking` command is `skip-external-locking` in MySQL version 4.*x* and above.

Generally, if you're using MySQL version 4.*x* and above, the `set-variable` directive is not required with the commands in this list.

```
set-variable=key_buffer=16K
```

This buffer is really small; if your database contains more than a few hundred lines of data in a text file, it would overload the capacity of this buffer. This may not overload the capacity of a text-based address book. If this is more than a database for personal use, you could reach this limit fairly quickly. In that case, you may want to consider the limits associated with one of the other sample configuration files.

```
set-variable=max_allowed_packet=1M
```

Naturally, the information associated with a database adds information over and above the actual data. By default, if it exceeds more than 1MB on a server, MySQL generates an error message.

```
set-variable=thread_stack=64K
```

This limits the stack size for each database thread. The default is sufficient for most applications.

```
set-variable=table_cache=4
```

You can limit the number of open tables in a database; smaller limits (the default is 64) are appropriate for smaller-scale databases.

```
set-variable=sort_buffer=64K
```

When processing a database, you may need additional buffer space in memory.

```
set-variable=net_buffer_length=2K
```

The MySQL server also reserves space for incoming requests, as defined by the `net_buffer_length`.

```
server-id=1
```

Generally, if you have a MySQL primary server, you should set its `server-id=1`; slave MySQL servers should have `server-id=2`.

```
[mysqldump]
```

You can transfer data between different types of SQL databases, as governed by commands under `[mysqldump]`.

```
quick
```

The `quick` option supports the dumping of larger database tables.

```
set-variable=max_allowed_packet=16M
```

The size of the `max_allowed_packet` for transferring tables to other databases, naturally, is larger than that for simple communication between the client and server.

```
[mysql]
no-auto-rehash
```

This stanza sets conditions for starting the MySQL service; in this case, `no-auto-rehash` makes sure this service starts more quickly.

```
[isamchk]
[myisamchk]
```

Relational databases such as SQL are processed by what is known as the Indexed Sequential Access Method (ISAM). The commands in these two stanzas are the same; they relate to the command of the same name, which checks and repairs database tables.

```
set-variable=key_buffer=8M
set-variable=sort_buffer=8M
```

You've seen these variables before with respect to the server. They're larger here to support a faster check and repair of the database.

```
[mysqlhotcopy]
interactive-timeout
```

During a database copy operation, as specified by `[mysqlhotcopy]`, connections can hang. The `interactive-timeout` variable by default sets the maximum time for a data transfer to 28,800 seconds (8 hours).

my-medium.cnf

The sample MySQL configuration file associated with medium-sized databases (`my-medium.cnf`) contains the same active stanzas as `my-small.cnf`. Under the `[mysqld]` stanza, the following commands support larger server databases:

```
set-variable=key_buffer=16M
set-variable=table_cache=64
set-variable=sort_buffer=512K
set-variable=net_buffer_length=8K
log-bin
```

Generally, the commands in this stanza support larger caches and buffer sizes on the server. We see a couple of new commands.

```
set-variable=myisam_sort_buffer_size=8M
log-bin
```

The `myisam_sort_buffer_size` command allows MySQL to index the database, and the second command supports binary logging.

```
[isamchk]
[myisamchk]
```

Naturally, the buffers are larger for database transfers under these stanzas. In both cases, this file includes the following commands, which sends and receives messages to and from the server.

```
set-variable=read_buffer=2M
set-variable=write_buffer=2M
```

my-large.cnf

The sample MySQL configuration file associated with larger databases (`my-large.cnf`) contains the same active stanzas as `my-small.cnf`. In this section, we'll compare the commands in `my-large.cnf` to the `my-medium.cnf` sample file. Under the `[mysqld]` stanza, the following commands support larger server databases:

```
set-variable=key_buffer=256M
set-variable=table_cache=256
set-variable=sort_buffer=1M
set-variable=myisam_sort_buffer_size=64M
set-variable=net_buffer_length=8K
```

There are three additional commands in this stanza. The `record_buffer` command saves scans for different tables in a database. The `thread_cache` command becomes useful with multiple requests; idle threads are cached, allowing new searches to take existing threads. As long as this keeps searches from starting new server processes, this can reduce the load on your system.

```
set-variable=record_buffer=1M
set-variable=thread_cache=8
set-variable=thread_concurrency=8
```

The `thread_concurrency` variable limits the number of threads that run simultaneously. The sample `my-large.cnf` file suggests that you should limit this to twice the number of CPUs on this computer; this particular setting corresponds to four CPUs.

my-huge.cnf

The `my-huge.cnf` file includes the same directives as in `my-large.cnf`. Naturally, the values assigned to most of the directives are larger and are suited for larger databases.

As described at `www.mysql.com`, organizations with substantial databases such as Google, Sabre, and NASA use MySQL. While we guess that these companies use commands other than what you'd see in `my-huge.cnf`, that at least gives you an idea of the power of the MySQL enterprise-level database.

Creating a Working Configuration

Check your Shadow Password Suiteconfiguration files, such as `/etc/passwd`. If you've installed the appropriate `mysql-server` RPM, you should find the user and group mysql in each of these files. Check the ownership of the key MySQL data and log files and directories. You'll see that the mysql user and group own these files and directories.

If you're just starting with MySQL, you can start with the default `/etc/my.cnf` configuration file. It should be sufficient for a small database with up to a few hundred entries. As your databases grow, you can copy the sample configuration files described earlier and overwrite your `/etc/my.cnf` file. As you gain skill with MySQL, you can modify the existing directives to suit your database and hardware.

Starting a MySQL Server

Before you can do anything more, you need to start the MySQL server. The process for starting the associated `mysqld` daemon is straightforward. Assuming you've installed with the RPMs included with Red Hat Enterprise Linux, you'll have similar start scripts as with other servers described in this book. The following commands start the MySQL server and then ensure that it starts the next time you boot Linux in runlevels 3 and 5.

```
# service mysqld start
# chkconfig --level 35 mysqld on
```

MySQL Users

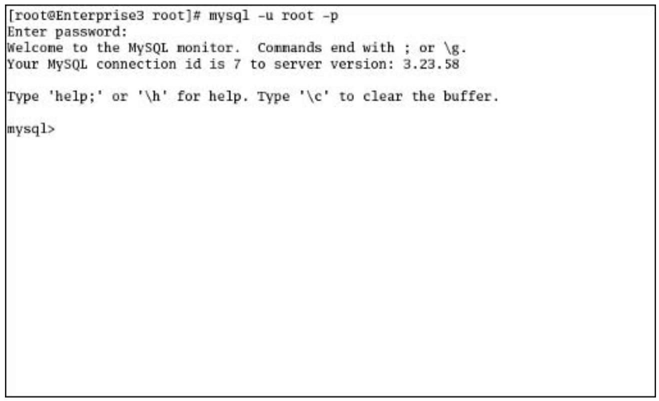
When you configure a MySQL server, you need to configure users on that server. These users are independent of the users and groups on your Linux computer. First, you'll need to add a root user for your MySQL system. Unfortunately, you'll need to add the password, in clear text, at the command-line interface.

For example, the following command creates a root user for your MySQL system:

```
# mysqladmin -u root password Ila451MS
```

Naturally, you'll want to create regular users to access your MySQL system. Next, you'll want to log into MySQL using the root account, as shown in Figure 26.2. Then you'll be on your way into the different world of MySQL commands.

FIGURE 26.2
Logging into MySQL

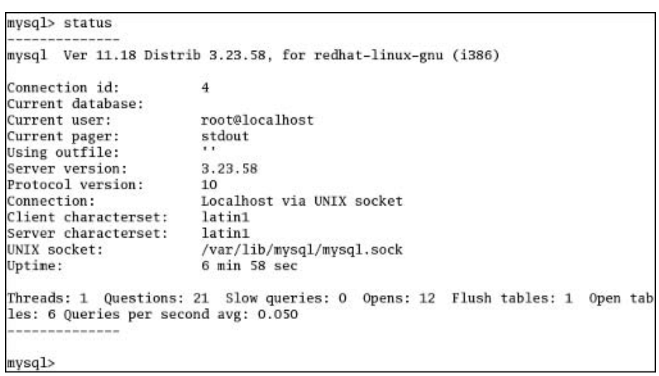


```
[root@Enterprise3 root]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 3.23.58

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

TIP *You should create MySQL users only when logged directly into the MySQL server. Unless you're using a secure protocol such as SSH, passwords are transmitted in clear text. And since MySQL passwords are typed directly on the command line, you should keep others from looking over your shoulder when you create MySQL users.*

There are a substantial number of MySQL commands. They may seem cryptic. There are many more than we can cover in this chapter. However, there are some essentials. For example, it's a good idea to check the status of your MySQL server. It's easy to do with the `status` command. The result is shown in Figure 26.3.

FIGURE 26.3
MySQL server status



```
mysql> status
--------------
mysql  Ver 11.18 Distrib 3.23.58, for redhat-linux-gnu (i386)

Connection id:          4
Current database:
Current user:           root@localhost
Current pager:          stdout
Using outfile:          ''
Server version:         3.23.58
Protocol version:       10
Connection:             Localhost via UNIX socket
Client characterset:    latin1
Server characterset:    latin1
UNIX socket:            /var/lib/mysql/mysql.sock
Uptime:                 6 min 58 sec

Threads: 1  Questions: 21  Slow queries: 0  Opens: 12  Flush tables: 1  Open tab
les: 6 Queries per second avg: 0.050
--------------

mysql>
```

NOTE *The MySQL documentation includes commands in uppercase. At the* `mysql>` *prompt, case does not matter, except for usernames and passwords.*

In the enterprise, presumably you'll have more than one user. You'll want to create users in the MySQL system. For example, if you want to create a user named DBguy, with all privileges on the default mysql database, you'd run the following command at the MySQL prompt:

```
mysql> grant all privileges on mysql.* to 'DBguy'@'enterprise3'
    -> identified by "Ila451MS"
```

This sets up DBguy on the local computer, with the password Ila451MS. That user can now log into your MySQL system with that password. You can confirm your user list with the following commands:

```
mysql> use mysql;
mysql> select * from db where db="mysql";
```

As you can see from Figure 26.4, the list of supported privileges is so large, it wraps around the standard text console. Now you can add the users who need privileges to the databases on your MySQL server.

FIGURE 26.4
MySQL use privileges

```
[root@Enterprise3 mysql]# mysql -u DBguy3 -h enterprise3 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25 to server version: 3.23.58

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from db where db="mysql";
+-------------+-------+--------+-------------+-------------+-------------+-------------+----------
---+------------+-------------+-----------------+------------+------------+
| Host        | Db    | User   | Select_priv | Insert_priv | Update_priv | Delete_priv | Create_pr
iv | Drop_priv | Grant_priv | References_priv | Index_priv | Alter_priv |
+-------------+-------+--------+-------------+-------------+-------------+-------------+----------
---+------------+-------------+-----------------+------------+------------+
| enterprise3 | mysql | DBguy3 | Y           | Y           | Y           | Y           | Y
   | Y         | N          | Y               | Y          | Y          |
+-------------+-------+--------+-------------+-------------+-------------+-------------+----------
---+------------+-------------+-----------------+------------+------------+
1 row in set (0.00 sec)

mysql>
```

It's possible that you want to grant limited privileges to certain users; for example, the following command at the `mysql>` prompt grants the noted privileges to the user DBguy3:

```
mysql> grant delete,insert,select,update on mysql.* to 'dbguy3'@'enterprise3'
```

Finally, if your DBguy user needs to log into your MySQL server, he'll need to run the following command. The switches are straightforward; the `-u` allows you to specify a username, the `-h` lets to note the computer with the database, and the `-p` makes the `mysql>` prompt for a password.

```
# mysql -u DBguy -h enterprise3 -p
Enter password:
```

Managing a MySQL Database

Now that you have a MySQL server and have created root and regular users, you're ready to create and manage a MySQL database. We've shown you how to log into the MySQL service. For the purpose of this chapter, I've used the commands in the previous section. First, I've used the `create database test` command to create a new `test` database. Then I've granted the MySQL user DBguy full access to that new `test` database.

Now let DBguy log into the MySQL database. At the `mysql>` prompt, you can point MySQL to the new database with the following command:

```
mysql> use test;
```

Now you can start creating a database.

Creating a Database

If the `use test;` command from the last section didn't work, you may have a problem as simple as a typo. It makes sense to check the name of the databases configured on your server, and that is easily done with the following command:

```
mysql> show databases;
+----------+
| Database |
+----------+
| dbase    |
| mysql    |
| test     |
+----------+
3 rows in set (0.01 sec)
```

Now you'll want to create the columns for your database table. The basic command is in a column name, column width, declare data format. For example, the following command sets up one column, entitled *date*, with 10 characters, with data input (`not null`):

```
date    varchar(10)    not null
```

To set up a database, you need more than one data point. The date can be a good place to start. But perhaps you'll want to break it down further. For example, let's take a company where you want to set up a database where every employee accounts for their time.

Logically, you may want to break it down into columns such as month, day, year, employee number, and time code. In this case, the time code represents the internal company code for a specific project or other miscellaneous duties. In that case, you could create a table with the following command:

```
mysql> create table timecode (
-> month      varchar(2)   not null,
-> day        varchar(2)   not null,
-> year       varchar(2)   not null,
-> emplnum    varchar(4)   not null,
-> code       varchar(4)   not null
-> );
```

Be careful with the syntax. Dashes aren't allowed in the column titles. Don't add a comma after the last column. If you have a problem, you may get a response such as ERROR 1064, with some cryptic message pointing to the syntax error. But if you don't get an error, you can check your work with the following command:

```
mysql> describe timecode;
```

Assuming this has the categories you need, you can start entering information into this database.

Adding Data

Now you can add information to your database. The standard at the `mysql>` command line is based on the `insert into` command. For example, if you wanted to add information for `emplnum` 1984 for March 20, 2005, you can run the following command:

```
mysql> insert into timecode
    -> values ('03','20','2005','1984','4321');
```

You can check the result with the following command, which allows you to view all data in the `timecode` database:

```
mysql> select * from timecode
```

Then you can add more data as needed. You can review the result from a set of data that we entered into the `timecode` database in Figure 26.5.

FIGURE 26.5
A Sample MySQL database

```
mysql> select * from timecode;
+-------+-----+------+---------+------+
| month | day | year | emplnum | code |
+-------+-----+------+---------+------+
| 03    | 20  | 2005 | 1984    | 4321 |
| 03    | 20  | 2005 | 1985    | 4321 |
| 03    | 20  | 2005 | 1986    | 4322 |
| 03    | 20  | 2005 | 1987    | 4322 |
| 03    | 20  | 2005 | 1988    | 4323 |
| 03    | 20  | 2005 | 1989    | 4324 |
+-------+-----+------+---------+------+
6 rows in set (0.00 sec)

mysql>
```

Loading Database Files

It's inconvenient to set up database access for all users. It's a lot easier to set up text files with the data that you can then load into the database. For example, you could set up a text file named `/tmp/timecards` with the data required for the database. You just need to make sure that each entry on a line is separated by a delimiter, entered with the Tab key. Based on the `timecode` database described earlier, I'd enter the first line as follows:

```
03<tab>20<tab>2005<tab>1984<tab>4321
```

Once you have the entries you want to add to your database, log back into your MySQL server. When you return to the mysql> prompt, enter the following command:

```
mysql> load data infile "/tmp/timecards" into table timecode;
```

You can confirm if your changes worked with the following command:

```
mysql> select * from timecode
```

Changing Data Entries

In any data entry operation, mistakes are made. Changes are needed. You can update and delete the entries of your choice. This assumes you've logged into MySQL and have run the appropriate use commands to call up the required database and table.

For example, assume your employee named bonds has an emplnum of 1 and a code of 755. The database is called homerun. The database entry may look like:

```
+-------+-----+------+---------+------+
| month | day | year | emplnum | code |
+-------+-----+------+---------+------+
| 04    | 10  | 2006 | 1       | 755  |
+-------+-----+------+---------+------+
```

Assume bonds' code number changes from 755 to 756. You can make the entry in the database from the MySQL prompt with the following command:

```
mysql> update homerun set code = "756" where emplnum = "1";
```

In the same database, assume you have an employee named jones with an emplnum of 1001. Since jones has moved to another organization, you'll want to delete his entry from the homerun database. You can do so with the following command:

```
mysql> delete from homerun where emplnum = "1001";
```

Summary

There are many databases which are designed to use the Structured Query Language (SQL). Two are available as part of Red Hat Enterprise Linux 3: MySQL and PostgreSQL. As MySQL is released under an open-source license, it is the preferred option of many Linux administrators and is therefore the database we cover in this chapter.

The MySQL server package is part of the Red Hat Enterprise Linux 3 Extras CD. It isn't included in the packages on the four standard binary installation CDs. You can download it using your account on the Red Hat Network or alternatively from the sites associated with the third-party "rebuilds," where the mysql-server package has been built using the source code that Red Hat has made publicly available.

The MySQL server package includes several sample configuration files that are suited for everything from small- to enterprise-scale databases. If you are administering a larger database, you'll need additional RAM and possibly additional CPUs. In that case, it's best to dedicate a system to the database.

Once you've configured and started a MySQL server, you'll want to create a separate set of users (including root) for that system. You can then create the databases of your choice and then finally enter information into those databases. You can do so at the `mysql>` command-line interface or with information organized in tab-delineated text files.

One of the demands on enterprise administrators is certification. Many companies are using certification as a basic qualification requirement for their jobs. In the next two chapters, you'll learn about available Linux certification programs, and you'll get an outline of what you need to learn to satisfy certification requirements and pass the associated exams.

Part 7

A Certification Primer

In this Part, you will learn:

Chapter 27

Generic Linux Certifications

Computer certifications are designed to measure some level of skill and or knowledge. While many people think certifications do not test "real" skills, companies still use them as benchmarks for hiring managers. Passing exams does help new users learn something about Red Hat Enterprise Linux. Certifications help people get real jobs. And people who are certified in Linux are to some extent advocates who can help promote others to adopt Linux.

As of this writing, three major generic Linux certification programs exist. CompTIA is revising the Linux+ exam for newer users with about 6–12 months of experience. Thomson owns the SAIR series of exams for Linux users with two or more years of experience. The Linux Professional Institute (LPI), a community organization, also has a series of exams for users with two or more years of experience.

This chapter briefly examines the requirements associated with the Linux+ exam, as well as the first-level exams from SAIR and LPI. While other Linux certification programs are available, in my opinion only Linux+, SAIR, LPI, and the Red Hat Certified Engineer/Red Hat Certified Technician programs covered in Chapter 28 have *currently* received significant industry and academic recognition.

This chapter provides a general overview of each of these generic certification programs and is not intended as a substitute for a preparation book or course for any exam. In this book, I do not cover areas that are normally beyond the scope of Red Hat Enterprise Linux 3, nor do I cover commands or configuration files that are now obsolete. However, you may need to know some of these topics, some of which may be obsolete, to answer questions on these exams.

Note *This chapter focuses on Linux certification exams, not Linux itself. For more information, some chapter references are provided.*

I have written books on several Linux certification programs, and you can use this book to help supplement your studies. This chapter covers the following topics:

- Preparing for the CompTIA Linux+ exam
- Studying for the LPI Level I exams
- Planning for the SAIR Linux Certified Administrator exams

Preparing for the CompTIA Linux+ Exam

As of this writing, CompTIA is revising the Linux+ exam for Linux users with about 6–12 months of experience. CompTIA developed this exam in concert with the people behind the LPI and SAIR exams; at least in theory, it fits in sequence as the entry-level Linux certification exam. Basic information about this exam is available from `www.comptia.org`.

The Linux+ exam focuses on the command-line interface. If you have any experience with Linux, GUI tools are usually distribution specific and many have come and gone, but key skills are related to the command-line interface.

As of this writing, CompTIA plans to have the revised exam available late in the fourth quarter of 2004. This means that if you know Red Hat Enterprise Linux 3 and follow the *new* CompTIA Linux+ exam objectives, you'll have an excellent chance of passing this exam.

NOTE *Don't underestimate the difficulty of the Linux+ exam. There are some who suggest that the first version of this exam was at least as difficult as what are supposed to be the mid-level Linux certification exams from LPI and SAIR. And older exam guides may not be so helpful; CompTIA has declared its intent to change 75 percent of the exam at the end of 2004.*

The Exam

Currently, the Linux+ exam consists of 94 multiple choice and/or multiple response questions. You have 90 minutes to answer all the questions. A passing score is 69 percent, or 65 questions. On the CompTIA scale for this exam, this corresponds to a score of 655.

Starting around the end of 2004, the new exam will include questions from the six subject domains listed in Table 27.1; when studying for this exam, pay attention to the relative weights of each subject.

NOTE *The Linux+ exam currently includes a 95th "question" asking for your permission to add your name to a database of certified professionals (assuming you pass). You can answer "no" without a penalty.*

TABLE 27.1: LINUX+ OBJECTIVE DOMAINS (2004 REVISION)

Subject	Weight on the Exam
Installation	19%
Configuration	20%
Management/maintenance	26%
Security	21%
Documentation	6%
Basic Linux hardware	8%

The final domain is somewhat surprising; it tests your knowledge of computer hardware, primarily at the "hands-on" level associated with CompTIA's A+ certification exam. This has reportedly dismayed a number of test candidates, which we believe is why CompTIA is planning to reduce its emphasis on PC hardware; the original Linux+ exam gave hardware 19 percent weight.

Some sections are essentially set up as lists because of the broad variety of topics covered by each of those domains.

TIP *The new Linux+ exam was still under development as of this writing. CompTIA may still make changes before this exam is released, possibly at the end of 2004. Refer to* `www.comptia.org/certification/linux/` *for the latest information.*

Installation

This Installation domain addresses what you should do when planning for and installing Linux. Related questions test your knowledge of hardware issues. You also need to know how to select a Linux distribution, available tools, licenses, and so forth.

This includes all the things you do on a PC before, during, and just after installing Linux. If you're not installing from a CD, you're probably installing from a networked server. There are a number of decisions common to all Linux distributions. In essence, this domain includes questions on the following issues:

1. List the hardware on the computer; make sure it's supported through research on available hardware compatibility lists.
2. Identify an installation method. If you're not installing from a CD, you may be using a boot disk. You could be installing Linux from a network server connected through FTP, NFS, HTTP, or even Samba. (Samba installations are not available for Red Hat Enterprise Linux 3.)
3. List the multimedia requirements for this computer; including video, sound, and more.
4. Identify the purpose of the computer. Are you installing Linux on this PC as a workstation, a desktop computer, or a server? If it'll be a server, will you configure it for a specific function such as applications, files, print management, mail, routing, or something else?
5. Add or subtract package groups depending on the purpose of your computer. If it's a workstation, you'll probably want to install the OpenOffice.org suite; if it's a dedicated mail server, you do not need to install the package groups related to Apache or DNS.
6. Plan how you're going to divide your hard disk into Linux partitions.
7. Assign a filesystem format to each partition, such as ext3, reiserfs, or swap.
8. Configure the bootloader, whether it is LILO, GRUB, or even ELILO (for 64-bit systems), and assign it to the Master Boot Record (MBR) or the first sector of a partition.
9. Know how to install or uninstall programs from RPM packages, DEB packages, and tarballs. (A DEB package is based on Debian Linux, with a package system similar but not identical to the RPM.)
10. Configure connections for your modem and network cards; manage the Internet Super Server, also known as `xinetd`. Some Linux distributions use the older `inetd` service.
11. Configure the basic parameters for the installation: language, keyboard, mouse, and time zones.

12. Set up various peripherals as needed, such as printers and scanners.
13. Set up users and groups with appropriate passwords and permissions. Keep it consistent with any security policies associated with your organization. Know the Shadow Password Suite, as well as SUID and SGID modes.
14. Configure the X Window. As of this writing, the main server is XFree86. However, as we describe in Chapter 29, this may change. Some distributions, including Fedora Linux, are adapting the X Server from `X.org`.

The objectives associated with the Installation domain are primarily addressed in Chapters 3 and 4.

Management/Maintenance

Management functions covered on the Linux+ exam can be broken down into five categories. User management allows you to configure users, groups, and file ownership. The Filesystem Hierarchy Standard (FHS) specifies how you organize different directories on available partitions. The startup and shutdown process determines the daemons and services that start when you boot Linux. There are a wide variety of management commands that even newer Linux administrators should know. Finally, commands can be collected into scripts to manage your systems when you want.

User Management

Administrative user management involves knowing how to manage, add, and delete users and groups. You need to understand the basic commands associated with this process, including `useradd` and `userdel`, as well as `groupadd` and `groupdel`. It helps to know how to edit the key configuration files, `/etc/passwd` and `/etc/group` directly, as well as how to make it work with the Shadow Password Suite of files and commands.

Managing users also means managing their ownership and permissions on different files and directories. You need to understand how to use `chmod`, `chusr`, and `chgrp` to revise default permissions set by umask. It helps to know how SUID and SGID modes work for different users and groups. Managing users accounts also includes setting quotas on files and space per user and group. Users need e-mail, which you can configure with basic sendmail, Postfix, or text mail clients. It's common to authenticate users on a network using NIS.

Ownership and permissions are covered in Chapter 6; user management commands and quotas are addressed in Chapter 9; mail servers and clients are addressed in Chapter 21; NIS is addressed in Chapter 23.

The Filesystem Hierarchy Standard

Linux is organized into directories based on the Filesystem Hierarchy Standard (FHS). It's important to know the types of files that you can find in at least the major Linux directories, including `/etc`, `/usr`, `/bin`, `/dev`, `/mnt`, and `/var`.

Creating additional files and directories requires an in-depth knowledge of some basic commands, including `cp`, `mv`, `rm`, `mkdir`, `rmdir`, and `ls`. There are dangerous variations on these commands that can easily delete everything on your hard disk.

Mounting of CDs, floppies, and shared network directories are managed manually with the `mount` and `umount` commands or automatically through `/etc/fstab`. Filesystems are created and managed with commands such as `fdisk`, `mkfs`, and `fsck`.

We cover the basic structure of the FHS and many of the associated commands in Chapters 6 and 7.

Startup and Shutdown

To answer questions about the boot process requires a basic knowledge of the associated configuration files, especially `/etc/inittab` and `/etc/fstab`. Many services are organized in different runlevels in `/etc/rc.d`; custom start parameters are often defined in `/etc/local`, and you can use a number of commands, such as `chkconfig`, to specify the runlevels associated with each service.

Each of these services can be controlled from scripts in the `/etc/rc.d/init.d` directory; with these scripts, you can at least start, stop, and restart the associated services. The default runlevel is set in `/etc/inittab`, and runlevels can be changed with `init`. Specific processes can be managed by PID number, with the help of a variety of commands. They range from `ps` and `kill` to `bg` and `fg`, and more.

The startup and shutdown process is addressed in detail in Chapter 11; process management is described in Chapter 13.

Other Management and Administration Commands

There are a number of other administrative skills that even a junior (Linux+) level administrator needs to know. For example, you should know how to use different remote access services, especially SSH, to manage computers on your LAN. Other skills you need for the Linux+ exam include the following:

- Be familiar with text commands, including `wc`, `cat`, `less`, `more`, `head`, and `tail`. Work with more complex commands such as `ln`; know how to combine commands using piping and redirection. Many of these commands are addressed in Chapters 6, 7, and 8.
- Monitor networks with basic commands such as `netstat`, `ping`, and `traceroute`.
- Know how to manage printers, print queues, and print jobs. While the Linux+ exam specifies LPD commands such as `lpq`, `lpc`, and `lprm`, we explain in Chapter 20 how CUPS can be administered with these commands.
- Be ready to repair and back up files, packages, and directories. Backups are common to removable media such as CDs and DVDs. Files can be repaired with appropriate `rpm` and `deb` commands.

Management Scripts

You can't be everywhere at once, and it helps to simplify administrative tasks with single scripts. Linux includes a number of preconfigured scripts that can help you administer a computer using the `cron` daemon; you can do the same tasks on a one-time basis with the `at` daemon.

Once you've learned the basic preconfigured scripts, you can create some of your own shell scripts to run on a schedule or as required; the `vi` editor is always available for this purpose. Common scripts include file manipulation commands such as `sed` and `awk`; as well as `find` and `grep`. These basic commands are described in Chapter 6; scripts are described in Chapter 13.

Configuration

Installation is usually not enough. You'll probably need to configure services and more to make sure Linux meets your needs. The Configuration domain addresses the configuration files associated with services, shells, hardware, and environment variables. This domain requires you to understand the following issues in some detail:

- Start basic TCP/IP network services using `/etc/sysconfig/network`, `/etc/named.conf`, and `/etc/dhcpd.conf`. Configure basic forwarding and routing parameters. Use `ifconfig` to set up a network card.
- Configure basic server services, including Samba, DHCP, DNS, and Apache.
- Set up automated mounting of local and network drives and partitions; configure drives in `/etc/fstab`.
- Get a DNS server and name resolution going by editing the appropriate configuration files, including `/etc/hosts`, `/etc/host.conf`, and `/etc/resolv.conf`.
- Know how to compile applications from tarballs; depending on your `Makefile`, certain `make` command switches support different configurations.
- Configure printers; you'll need a basic knowledge of CUPS and the Samba configuration file `smb.conf` stanza known as `[printers]`.
- Customize the syslog daemon and related log configuration directives for different error levels and file locations.
- Set up terminal emulation in the X Window.
- Customize environment variables, such as those to look through different directories (PATH), to locate the GUI (DISPLAY), and to specify the type of command-line terminal (TERM).

The objectives associated with this domain are addressed in a wide variety of chapters. Remember, the Linux+ exam is a near-entry-level Linux exam, so you won't need to know how to configure these services in too much detail.

Security

Security is one of the new domains under development for the Linux+ exam. This exam tests your knowledge in practical areas, such as whether you may want to keep your servers or network equipment physically secure. It also tests your knowledge of some detailed security tools. This domain addresses the following topics:

- Configure basic security files. This includes `/etc/hosts.allow` and `/etc/hosts.deny` for network security, `/etc/sudoers` for root user security, `/etc/ftpusers` for security with WU-FTP, and `sshd_config` for the SSH service.
- Customize security at the user level, including accounts, logins, quotas, and more. Grant root privileges to appropriate users. Set up password policies as appropriate.

- Set up a firewall to protect the computer and network; know basic `iptables` commands to limit access through common TCP/IP ports. Tools such as Snort and PortSentry can help detect break-ins to your network.
- Manage services for security. With the right `rpm` and Tripwire commands, you can check for alterations. Appropriate process permissions with SUID and GUID settings allow regular users into the right services.
- Use encryption. Different algorithms are available, including blowfish, 3DES, MD5, and more.
- Review general and service-specific log files to help detect problems.

We cover security measures in detail in Chapter 22. Log files for different services are associated with appropriate chapters.

Documentation

Documentation is the other new domain under development for the Linux+ exam. For the most part, this domain is related to documenting the status of the system. Only a minor part addresses online documentation. You'll need to know the following topics:

- Set up a performance baseline. Know and record what happens to your Linux system under normal conditions.
- Record the installed configuration, including a record of packages, network settings, key configuration files, and more.
- Write procedures for basic administration activities, including installation, configuration, security, and administrative management.
- Use system and application service log files to troubleshoot problems.
- Refer to local and online documentation as needed, from man pages to `README` files and HOWTO documents.

Basic Linux Hardware

This is one area of the Linux+ exam where some of the topics go well beyond the Linux operating system. While there are important Linux commands and files related to hardware configuration, there is a significant body of material from CompTIA's A+ hardware exam.

The new Linux+ exam is significantly reducing the emphasis on hardware from 19 percent to 8 percent of the total exam.

Linux Hardware Tools

Historically, PC hardware has been a problem for the Linux operating system. Some manufacturers still build their hardware to support only Microsoft Windows. For the Linux+ exam, you do need to know how it supports hardware: in drivers, in `/proc`, and during the boot process.

Hardware drivers are part of the `/dev` directory, and their modules are stored in kernel directories. If you're having problems with hardware, your problem may fall into one of these categories:

- The device may not be supported, which you can check through the resources cited in Chapter 2.
- The device driver may not be connected, which you can check with the `lsmod` command. Alternatively, you can try `modprobe` and `kudzu` if automatic detection doesn't work. Otherwise, you may need to recompile your kernel to add appropriate support.
- There may be a conflict. Devices that are properly linked to the kernel are shown in one or more files in the `/proc` directory. Deductive reasoning will then show you what hardware lost out because of the conflict.

For more information on hardware detection, see Chapter 2. The `/proc` directory is a virtual directory that contains hardware settings organized in descriptive files. We discussed this directory in Chapter 11.

The Linux+ exam does require that you know how to troubleshoot hardware. The aforementioned `lsmod` and `kudzu` tools enable you to diagnose driver problems. Files in the `/proc` directory help you figure out what was not detected. As discussed in Chapter 11, you can find boot messages by issuing the `dmesg` command or by checking the latest `/var/log/messages` file.

Finally, the Knoppix distribution, which can be loaded directly from a CD, is gaining acceptance as an alternative method for rescuing Linux and even Microsoft Windows systems from crashes.

Non-Linux Hardware Issues

Part of the Linux+ exam is dedicated to hardware issues that are beyond Linux. If you're comfortable with CompTIA's A+ hardware exam, you're probably in good shape here.

While some of these issues are covered in Chapter 2, you need more information for this part of the Linux+ exam. For that purpose, I recommend the following book: *The Complete PC Upgrade and Maintenance Guide,* 15th edition, by Mark Minasi (Sybex, 2004). You'll also find more information on each of these acronyms in that book. The non-Linux hardware issues on the Linux+ exam include the following:

- Know the look and feel of the hardware that connects floppy, IDE, and SCSI drives to your computer. It helps to know the kinds of cables and pins associated with each of these types of drives. Know the SCSI numbering system.
- Other internal PC hardware conforms to the ISA, PCI, and AGP standards; know how they are connected to RAM memory.
- Hot-swappable hardware conforms to the USB, IEEE1394, and PCMCIA standards.
- Hardware connections are organized mostly into IRQ ports, I/O addresses, and DMA channels. Know the IRQ ports and I/O addresses associated with the first four COM and the first two LPT ports.
- Know how to work with APM (Advanced Power Management) and ACPI (Advanced Configuration and Power Interface).

Studying for the LPI Level I Exams

The Linux Professional Institute (LPI) is a community-based, not-for-profit group organized solely to create a generic certification for Linux administrators. Although there are three levels of LPI Certification (LPIC), the following sections address only the requirements associated with the two LPI Level 1 exams. While the LPI Level II exams are available, I believe that many who want a higher level Linux certification will choose to work toward becoming a Red Hat Certified Engineer (RHCE) or a Red Hat Certified Technician (RHCT), which is the subject of Chapter 28. As of this writing, the LPI Level 3 exams have not been developed.

NOTE *One of LPI's past objectives for its Level I exams is to measure knowledge at the "prerequisites" level for the RHCE. They use multiple choice questions, instead of the hands-on exercises associated with the Red Hat exams.*

According to LPI, administrators who pass its Level 1 exams are suitable as junior Linux system administrators. They have the skills at the command-line interface and can perform basic maintenance of users, back up and restore systems, and run Linux through the boot cycles.

The LPI Level I exams are known at various testing centers as the General Linux I and General Linux II exams. LPI has recently implemented Release 2 of these exams. To qualify as an LPIC-1, you need to pass both General Linux exams. The objectives for each exam are often under development; for the latest information, refer to `www.lpi.org`.

As the LPI exams are explicitly "distribution neutral," expect a near-exclusive focus on the command-line interface. Linux GUI tools are by and large distribution specific.

General Linux I

The LPI General Linux I exam includes five topics, each of which includes a number of different objectives. Unlike the Linux+ exam, the Hardware and Architecture objective tests your knowledge of hardware configuration within Linux. Linux Installation and Package Management addresses the way you organize a hard disk as well as how you manage Linux packages. An extensive knowledge of various GNU operational and administrative commands, as well as the `vi` editor, is required. You need to know how to manage Linux devices and filesystems. And finally, you're tested on your knowledge of X Window configuration.

A New Certification

With its purchase of SUSE, Novell has made a serious commitment to Linux. Novell's SUSE Enterprise Linux Server 9 is certainly a serious competitor to Red Hat Enterprise Linux 3. As a part of that commitment, Novell has developed its own certification at the end of 2003, the Novell Certified Linux Engineer (CLE). While the LPI Level I exams are a suggested prerequisite, the CLE hands-on exam requires in-depth knowledge of Novell Nterprise Linux Services, Novell's eDirectory, and Virtual Office. If Novell is successful in expanding market share for the SUSE distribution, it's reasonable to expect that CLE will become an important Linux certification. These are all tools and applications that go beyond Linux, so we do not cover it here.

The requirements associated with the LPI Level I exams are constantly evolving. When I participated in the revision process several years ago, the participatory nature of the process meant that the exam coverage is constantly expanding. The following is just an overview, and when the people behind the LPI exam revise it again, I suspect this overview will be incomplete.

Hardware and Architecture

For this part of the exam, you need to know how Linux works with basic PC hardware components. The hardware-detection process starts with the BIOS and continues with memory, hard drives, and expansion cards. Special attention is paid to other peripherals, including modems, sound and network cards, and USB devices. Most of these topics are addressed in Chapter 2. You may need to go further, as it helps to know the right IRQ, I/O, and DMA settings for different BIOS configured ports.

You can find information on detected hardware in the `/proc` directory. As described in Chapter 11, this directory includes a number of files that document detected hardware directly and through different channels, such as IRQ ports, through `/proc/interrupts`.

Modems and sound cards often present special challenges. As discussed in Chapter 2, Winmodems are designed to use Microsoft Windows driver libraries. Sound cards can require multiple DMA addresses, which can be difficult to detect. However, Linux is working seamlessly with an increasing number of Winmodems and sound cards. Tools such as `minicom` (see Chapter 16) and `redhat-config-sound` (see Chapter 2), formerly `sndconfig`, to help you configure these components.

With the accelerating use of higher-speed connections to the Internet, you need to know how to configure ISDN adapters, as well as DSL/Cable modem connections. While Red Hat uses the Internet Configuration Wizard, you need to know how to make these changes at the command line and with the appropriate text files. All are described in Chapter 16.

Using SCSI devices requires a basic understanding of the SCSI BIOS, available SCSI hardware, and ID numbers; these components are documented in the `/proc/scsi` directory.

Linux administrators may work with older ISA and PCI cards, and should know how to prevent IRQ, DMA, and I/O conflicts between these cards. The LPI exam as of this writing cites commands that are obsolete for Red Hat Enterprise Linux 3, including `pnpdump` and `isapnp`.

Linux support for USB is good; however, it is still under development for newer devices, especially those that conform to the USB 2.0 standard. In this area, you should learn about the `lspci` and `usbmodules` commands as well as the configuration files in the `/etc/hotplug` directory.

Linux Installation and Package Management

In Linux, you can install programs during the installation process, or afterward with commands such as `rpm`. During the installation process, you should assign different filesystems to different partitions. You also need to configure a bootloader. You can install different groups of programs; some may require special program libraries.

The basics of partitioning a hard disk layout are covered in Chapter 2. The different filesystems, such as root (`/`) and `/boot`, are explained in Chapter 7. The discussion includes coverage of other filesystems such as swap space, as well as `/var` and `/home` as mount points. Older computers require `/boot` on a hard disk cylinder below 1024; otherwise, your BIOS may not be able to find it to boot Linux.

While Red Hat Enterprise Linux has moved toward making GRUB the default bootloader, several other distributions still use LILO. You need to know the files and commands associated with both bootloaders.

Some of the guts of Linux are the shared libraries. Most major Linux libraries are stored in directories defined in `/etc/ld.so.conf`. They're managed through commands such as `ldd` and `ldconfig`.

You need to know how to install programs from source code in a tarball; the best example is described in Chapter 12 (which examines the Linux kernel). As discussed in Chapter 10, the standard Red Hat installation process uses RPM packages, but you also need to know the Debian `dpkg` system and the associated commands, including `dselect`, `apt-get`, and `alien`.

GNU and Unix Commands

This objective is straightforward. It encompasses most commands, utilities, scripts, and so forth that you may run at the command-line interface.

One group of commands allows you to navigate directories and create files. Another group lets you set the parameters associated with the shell. In addition, there are text stream commands that you use to process files in different ways. Directional commands and arrows redirect standard input, standard output, and standard error.

Search commands can drill into and return selected data from different text files. In addition, this exam tests your knowledge of the `vi` editor, including basic commands associated with Command, Insert, and Execute modes. Most of these bash and related environment commands are covered in Chapters 6, 7, and 8.

Linux is a multitasking system that always includes a number of running processes. Some commands let you create, monitor, and kill running processes. Others help you reprioritize how these processes compete for time from your computer's resources. Chapter 13 covers these process management commands.

Devices, Linux Filesystems, Filesystem Hierarchy Standard

Hard disks can be divided into partitions. Each partition is associated with a Linux device file. Once they're formatted and configured, you can mount a specific directory filesystem to each partition, and you can document the changes in `/etc/fstab`. Quotas can be set in different partitions; permissions can be managed.

Finally, there are a number of commands that you use to search for files. Some, like the `find` command, search in real time; others, such as the `locate` command, are faster because they use static, but possibly obsolete, databases.

A couple of key commands are associated with creating and then formatting a partition: `fdisk` and `mkfs`. You use these utilities to configure different partition types and format to different filesystems, such as ext3 and vfat.

Several other commands allow you to maintain the integrity of these filesystems. The `du` and `df` commands relate to disk space usage. The `fsck` command checks the integrity of a partition; related commands include other important integrity data for that partition.

Once partitions are documented in `/etc/fstab`, they are relatively easy to mount and unmount. It's important to know the syntax of `/etc/fstab`.

Quotas can be configured by mounted partition, documented in `/etc/fstab`, and activated with a number of different commands. You can create quotas for users and/or groups.

File permissions and ownership can be managed with several key commands: `chmod`, `umask`, `chown`, `chgrp`, and `chattr`. The `chattr` command sets the immutable flag, which prevents accidental deletion even by the root user.

Files are commonly linked. Device files such as `/dev/modem` are linked to their actual ports (such as `/dev/ttyS0`) for ease of identification, using the `ln` command.

You want to be able to search through files. Commands such as `find` search directly in real time; commands such as `locate` use a regularly updated database created by `updatedb` and are scheduled as a `cron` job.

You can learn more about these commands in Chapters 6–9.

The X Window System

Despite the number of Linux administrators who prefer to work at the command-line interface, LPI recognizes that it is important to know how to configure the X Window. Users demand it. The LPI General Linux I exam divides this process into three parts: basic configuration, the setup of a display manager, and the configuration of the window manager environment. These topics are covered in Chapter 29.

While Red Hat Enterprise Linux has incorporated its own `redhat-config-xfree86` configuration tool, the more generic `xf86config` utility should still be usable on other Linux distributions. In all cases, your objective is to configure the `/etc/X11/XF86Config` file to set the basic parameters for the video card, monitor, fonts, keyboard, and mouse.

Note *Like Red Hat Enterprise Linux 3, the LPI exam has not been updated (as of this writing) for the changes being made away from the XFree86 servers. However, we believe that LPI may make significant changes in this area in its next revision.*

The display manager is the graphical login screen you configure for your system. This assumes that you revise `/etc/inittab` to start Linux in the GUI runlevel—5 for Red Hat Enterprise Linux (it may vary with other distributions; remember, the LPI exam is distribution neutral). Three basic options are available for display managers: XDM, GDM, and KDM.

When you configure a desktop environment such as GNOME or KDE, you need to be able to customize it. It's common practice to set up a desktop environment to start with the same "look and feel," which would include common icons and perhaps an X terminal. Configuration changes are normally documented in hidden files in users' directories.

General Linux II

The LPI General Linux II exam is divided into nine different topics. Each topic includes several objectives. These topics fall into a broad variety of categories, including the kernel; boot, initialization, shutdown, and runlevels; printing; documentation; shells, scripts, programming, and compiling; administrative tasks; network fundamentals; networking services; and security.

The Kernel

The Linux kernel is at the core of the operating system. It helps Linux talk directly to your hardware. Kernel modules connect many components, including peripherals, to your operating system. Chapter 11 describes a number of commands that help you manage kernels and their associated modules. Some modules are loaded during the boot process from `/etc/modules.conf`.

If the desired module isn't available, you may have to revise and then recompile the Linux kernel. You'll need to know the basics of the process for the General Linux II exam. This includes the configuration utilities, as well as the commands associated with cleaning, compiling, and modularizing the kernel. This complex process is detailed in Chapter 12.

Boot, Initialization, Shutdown, and Runlevels

This topic is more straightforward than the title. In essence, you need to know how to manage Linux during the basic startup and shutdown process.

When your computer boots, it normally points to the Master Boot Record of a hard disk, where it finds a bootloader such as GRUB or LILO. Once you select Linux, the kernel loads and then starts the `init` process, which initializes hardware, loading kernel modules noted in `/etc/modules.conf`. If you need to see what happened, you can inspect the startup by issuing the `dmesg` command or by viewing the appropriate part of the `/var/log/messages` file.

Linux starts in a specific runlevel with a group of services defined in the `/etc/rc.d` directory. You don't always have to operate in a specific runlevel; the `init` command can change runlevels, even to reboot or shut down your computer. You can modify the `id` command in `/etc/inittab` to change the default boot runlevel. These processes are described in Chapter 11.

Printing

There are two major print services: LPD and CUPS. While CUPS is the current default for Red Hat Enterprise Linux, it is not the default for all Linux distributions. In either case, the basic administrative tasks are the same.

You must configure local and remote printers. Whether you use CUPS or LPD, printers are listed in `/etc/printcap`. Printers use different filters; two are aps and magicfilter.

Once configured, you have to manage any printers that you have installed, along with their queues. With the `cups-lpd` RPM, you can do so on both services with commands such as `lpc`, `lpq`, and `lprm`. Within each queue, you may need to manipulate individual print jobs. The CUPS service is covered in Chapter 20.

Documentation

Linux documentation seems to be available everywhere. You must know where the documentation is located on your computer, where the major sources are on the Internet, and how to send document-related messages to your users.

The most basic Linux documentation is based on the man pages, which are available for most commands and many configuration files. You can search through man page titles with `apropos`. Documentation associated with a number of services can be found in various `/usr/share/doc` directories.

You can also find a number of sources of Linux documentation online. Perhaps the most prominent of these is sponsored by the Linux Documentation Project at `www.tldp.org`. The home page associated with most services typically includes links to extensive documentation. See Appendix A for more sources.

You can configure a number of logon messages for your users. By default, these messages are configured in `/etc/motd`, `/etc/issue`, and `/etc/issue.net`.

Shells, Scripting, Programming, and Compiling

Administrators create scripts to simplify their lives; for example, you can configure scripts to run jobs in the middle of the night automatically. The most common scripts are described in Chapter 13. Basic scripts associated with the login process are also available. With the right permissions, you can create and activate your own scripts as well.

When you customize your shell, you can add commands to startup scripts in your directory. If you use bash, the scripts are `~/.bash*`. You can also set variables with a number of different commands, such as `set`, `unset`, and `export`. This process is described in Chapter 12.

Administrative Tasks

Linux administrators have several important tasks. They manage user and group accounts. They tune the user and system environment. They customize log files for special needs. They automate repetitive and untimely tasks. They back up and restore data. And they synchronize the time on their servers.

For the General Linux II exam, you need to know the commands and files associated with maintaining users and groups in the Shadow Password Suite described in Chapter 9.

Because users customize their own environments, you may want to create a default environment in `/etc/profile` and `/etc/skel`, which is covered in Chapters 9 and 12.

Log files are configured in `/etc/syslog.conf` and stored in sequence in the `/var/log` directory. The `logrotate cron` job ensures that these files are manageable. System logs are discussed in Chapter 13.

As a Linux administrator, you need to run jobs that can load a system. Since you don't want to affect network performance and upset your users, you should run jobs such as data backup in the middle of the night. These jobs can be automated with the `cron` and `at` daemons, discussed in Chapter 13.

One important administrative task is the creation of backups. Basic backup strategies are covered in Chapter 14. If you have multiple servers, it is important to keep their clocks synchronized. This is possible with the Network Time Protocol (NTP); you can configure the NTP daemon through the `redhat-config-time` tool described in Chapter 13.

Networking Fundamentals

Every computer administrator needs to know the fundamentals of networking. The basics of TCP/IP are covered in Chapter 15, where you can learn about IP addressing, CIDR notation, and the TCP/IP ports associated with basic services (you may actually have to memorize a few port numbers). A number of other network fundamentals are covered in other chapters.

When you configure TCP/IP on your computer, you're editing a number of basic configuration files. Even though Red Hat consolidates a lot of this data in the `/etc/sysconfig` directory, the basic configuration files such as `hosts`, `resolv.conf`, and `host.conf` are still in the `/etc` directory. You can use TCP/IP configuration commands to set IP address and other parameters.

It's important to be able to configure a Linux workstation as a Point-to-Point Protocol (PPP) client, so users can connect to their ISPs via a telephone modem. You can learn more about this process by reading about `minicom` in Chapter 16.

Networking Services

The LPI exam requires some knowledge of network services, including `xinetd`, sendmail, Apache, NFS, Samba, DNS, and SSH. We address configuration of these services in Chapters 18, 19, 21, 22, 24, and 25. One common element in these chapters is learning how to configure each of these services to start the next time you boot Linux. When you take the LPI exam, you should recognize that configuration file locations may vary by Linux distribution.

The `xinetd` service is the successor to the `inetd` super server. While `inetd` is obsolete for Red Hat Enterprise Linux 3, it is used on other Linux distributions. However, you can still protect individual services through `/etc/hosts.allow` and `/etc/hosts.deny`. For the LPI exam, you need to be able to protect these services by name and host. The `xinetd` service is covered in Chapter 18.

The most common outgoing e-mail server is sendmail. The location of some sendmail configuration files in Red Hat Enterprise Linux 3 may be different from other Linux distributions. Know how to configure aliases, especially when you want to forward mail to other users. We cover sendmail in Chapter 21.

Apache is the most common web server on the Internet, and LPI covers basic Apache configuration. You need to know enough to get it going; however, custom configuration is an advanced skill for other exams such as the RHCE. Chapter 25 examines the Apache web server.

Because NFS and Samba are both designed to share directories on a network, their configuration is grouped as one objective on the LPI exam. You need to know the basics of configuring NFS and Samba to share directories with other computers on your LAN. Microsoft Windows configuration issues, such as domain management, are explicitly excluded from the LPI exam. We cover NFS and Samba in Chapters 22 and 24.

As your network grows, you may need to configure a DNS server on your LAN. Under this objective, the General Linux II exam covers the basic DNS and host configuration files, including `/etc/hosts`, `/etc/resolv.conf`, and `/etc/named.conf`. DNS configuration is covered in Chapter 19.

One way to set up a secure connection to a remote computer is with the SSH service. You need to know how to configure SSH with the appropriate private and public encryption keys, as well as how to block connection attempts from unwanted users. SSH configuration is covered in Chapter 18.

Security

There are several steps you can take to secure your Linux system. The General Linux II exam includes a number of skills in this area, including the configuration of TCP Wrappers, `ipchains`, and `iptables`. We cover these skills in Chapters 17 and 18. Other security measures are more generic, such as upgrading and verifying RPM packages.

Some security measures are available for each computer. You need to know how the files of the Shadow Password Suite help protect your system. It's also a good practice to deactivate and even uninstall unused network services.

Security also relates to user accounts. You can limit the damage if a cracker finds someone's password. Quotas limit the amount of space assigned to a user or a group. User accounts can be limited

in scope; for example, you can set passwords to expire after a certain time. These options are described in Chapter 9.

Planning for the SAIR Linux Certified Administrator Exams

The last major Linux certification program we'll present in this chapter was developed at the University of Mississippi by Dr. Tobin Maginnis. The SAIR (Software Architecture Implementation and Realization) exams were also developed to measure the knowledge of more experienced Linux users. They suggest that candidates should have at least two years of experience.

While there are three levels of SAIR Certification, this section addresses only the requirements associated with the four SAIR Linux Certified Administrator (LCA–Level 1) exams. While the SAIR Linux Certified Engineer (Level 2) exams are available, I believe that many who want a higher level Linux certification will choose to work toward becoming a Red Hat Certified Engineer (RHCE) or Red Hat Certified Technician (RHCT), discussed in Chapter 28. As of this writing, the SAIR Master Linux Certified Engineer (Level 3) exams have not been released.

There are four SAIR LCA exams: Installation and Configuration; System Administration; Networking; and Security, Ethics, and Privacy. If you pass one of the first two exams, you qualify as a Linux Certified Professional (LCP). You need to pass all four of these exams to become a Linux Certified Administrator (LCA). You can learn more about the SAIR exams from its website at `www.linuxcertification.org`.

The SAIR exams are much more focused on the academic market. Some of the exams go beyond what we cover in this book. These exams are backed by Thomson/Prometric and the Linux Professional Group, which is using its power to promote this certification. As these exams are also "distribution neutral," there is a focus on the command-line interface. These exams are also evolving, so check the SAIR Website for the latest requirements.

The objectives for each exam are divided into six areas: theory of operation, base systems, shells and commands, system services, applications, and troubleshooting.

NOTE *Although SAIR and LPI test Linux users in the same market, their supporters are critical of each other's methods. If you take one or the other of these exams, be aware that you could walk into a debate nearly as vigorous as that between the proponents of Linux and Microsoft Windows. But peace is possible. While I've written a book on the SAIR Installation and Configuration Exam, I've also helped revise the objectives for the LPI exam.*

Installation and Configuration

The SAIR Linux/GNU Installation and Configuration Exam is in a way an overview of the other SAIR exams. To pass this exam, you need the skills required for the other three LCA exams. You're not just installing Linux—you're configuring various Linux services, at least on a basic level.

Theory of Operation

One controversial part of the SAIR exams is its focus on Linux history and licenses, especially the GPL and the open-source licenses. The belief is that a solid understanding of these licenses is needed for anyone who wants to convince his or her management to adapt Linux on their PCs. The basics of these licenses and Linux history are covered in Chapter 1; you can read the GPL in Appendix B.

More conventionally, this exam also tests your skills with PC hardware. You have to be familiar enough with a PC to know what you can configure with Linux and to realize the risks that may be involved with key components such as monitors. You also need to know the basics of Linux partitions using tools such as `fips` and `fdisk`. These skills are covered in Chapters 2 and 3.

Questions associated with this section also test your knowledge of the basic components of Linux: the kernel, init, daemons, network configuration, basic processes such as shells, and the X Window. You need a lot of this information to configure Linux during the installation process; we address this topic early in this book, in Chapters 1, 3, and 4.

Base Systems

This section addresses what you do during and just after the basic Linux installation process. Because SAIR is also a distribution-neutral exam, it takes commonalities from multiple distributions such as SCO, SUSE, Mandrake, Debian, and Slackware. Each of these distributions has strengths and weaknesses, and this exam compares them on a sufficiently broad level that the lessons are still true today.

You can install Linux from a wide variety of media, locally from CDs, or over a network. Older versions of some Linux distributions could even be installed solely from floppy disks. The exam addresses basic hardware installation difficulties, such as printers, video adapters, and Winmodems.

The exam asks you to recognize the basic steps common to all Linux distributions. These include hard disk partition planning, with provisions for dual-boot installation, and swap space. Once you boot the installation kernel, you create and format the desired partitions. Next, you select from a menu of packages to install, configure the bootloader, and then configure the X Window system. SAIR refers only to the LILO bootloader. To this point, you can learn about the topics addressed on the exam in Chapters 2–5.

This exam also tests your knowledge of the basic startup and shutdown process once Linux is installed. Many distributions set up `/etc/inittab`, `/etc/fstab`, and service startup scripts differently. The Red Hat method, which is a little different from other distributions, is described in Chapter 11.

You'll also need to know the Filesystem Hierarchy Standard (FHS), along with the commands used to `mount` and `umount` partitions on specific directories. There are special filesystems such as `/dev` and `/proc`, which are abstractions. We describe the FHS in some detail in Chapter 7.

This exam tests your knowledge of the tools used to inspect the current state of the system, including `ps` and `who`. Understand file and directory permissions, and know how to create user accounts in different ways. These topics are addressed in Chapters 9 and 13.

You need to know some of the basics of the X Window: how it starts and the associated configuration files, fonts, and other basic actions. These concepts are covered in the KDE and GNOME Chapter 30.

Finally, you need to know how to set up user accounts from the command-line interface. You need knowledge of some of the GUI tools for creating users, such as Linuxconf, YaST, and Lisa. YaST is the only one of the three tools that is still in use on modern Linux distributions.

You also need to know about the basic commands related to adding and deleting users. It's helpful to know how to add users by directly editing `/etc/passwd` and `/etc/group`. These techniques are covered in Chapter 9.

Shells and Commands

This section is fairly straightforward. For the Installation and Configuration exam, you need to know the basic configuration files and commands that set up a user's shell and environment variables. And you must know the basic commands for navigating and searching through files and directories on the Linux operating system.

The basic shell configuration files are described in Chapter 8, which includes generic files in `/etc` and user-specific files in each user's home directory. There are a number of key environment files that you can see with the `env` command and set from shell variables with the `export` command.

The basic commands covered help you navigate around directories, create new files and directories, show disk and partition usage, `locate` and `find` files, manage tarballs and other compressed packages, and find characteristics within text files with commands like `grep` and `wc`. Chapter 6-8 describes most of these commands.

System Services

For the system services section of the Installation and Configuration exam, you need to understand printer configuration, window managers, and the X Window architecture. Red Hat Enterprise Linux uses CUPS as the default print server; you may need to know LPD in some detail for this exam. The latest Linux distributions are replacing XFree86 with the X server. Yet this exam is focused on XFree86, which is still a part of Red Hat Enterprise Linux.

With respect to the older LPD print service, you'll need to know several of the arcane commands associated with `/etc/printcap`, as well as the functionality of LPD-related commands such as `lpr`, `lpq`, `lpc`, and `lprm`.

As for X Window configuration, this exam includes references to obsolete tools such as `Xconfigurator` and `XF86Setup`. The exam was written before the XFree86 4.*x* server was released. However, the basic principles of what you configure in `/etc/X11/XF86Config` remain the same and are addressed in Chapter 29.

Although GNOME and KDE are the most popular desktop environments for Linux, at least a dozen alternatives are available. For the exam, you need to know that there are alternatives such as AfterStep, Window Maker, fvwm95, Enlightenment, and Blackbox.

When you add and remove hardware, Linux often adds the associated drivers automatically. However, Linux does have occasional hardware problems; you can manage driver modules with commands such as `insmod`, `rmmod`, and `modprobe`.

Applications

This section can be divided into two areas: documentation and applications. Linux includes extensive documentation on your computer in `rom` pages, `info` commands, and `README` files. Linux also includes extensive online documentation from sources such as the LDP, Linux Today, and more; these are described in Appendix A.

You don't have to know Linux applications in any detail for this exam. It's enough to know what is available. For example, as long as you know about gFTP, Mozilla, Netscape, Telnet, WordPerfect, StarOffice, Applixware, Ispell, The GIMP, X-Fig, and ImageMagick, that should be sufficient.

One critical detail is the difference between text-processing utilities such as `vi` and binary word processors such as the OpenOffice.org writer. If you save configuration files with the OpenOffice.org writer, they'll probably be in binary format, and Linux won't be able to read them.

Troubleshooting

The skills you need for troubleshooting on this exam are quite varied, and many of them are covered in different chapters in this book. Some of these topics include the following:

- Installation problems related to bad media such as a CD or boot floppy result in "read" or "file not found" errors.
- Archive errors related to bad downloads of tarballs lead to bad format errors.
- During the installation process, if a partition gets full, you may not find it unless you check one of the other installation consoles.
- If the boot process stops with `LI`, the kernel or bootloader may be missing from the `/boot` directory.
- Bad block errors may indicate a physical problem with your drive; it's sometimes addressed by the `badblocks` command.
- Rescue disk mode helps you recover from errors such as a corrupt MBR, missing `/etc/passwd` file, or a lost dynamic library that you may have to reinstall.
- When programs are locked, you need to know how to `kill` the associated processes from another virtual console.
- Printer problems are similar to network problems; most issues are with physical connections. Otherwise, check log files, spools, and configuration files. Test the printer with text commands. Remember, this exam is based on LPD, and Red Hat uses CUPS as the default.
- If you're collecting troubleshooting data, look through the log files. Most are stored in the `/var/log` directory.

System Administration

Functionally, there is significant overlap between this exam and Installation and Configuration. In essence, the topics covered on this exam include more advanced system administration techniques.

Theory of Operation

The techniques in this area are at a relatively high level, as befits the title. You need to know how inodes work. You need to know the fundamentals of filesystems and the FHS. `cron` jobs are important, as are backups and RAID. Good administrators monitor and tune their systems to optimize performance. And good administrators are always ready with a rescue disk to recover from system disasters.

Although most Linux distributions have moved to journaling filesystems such as ext3, most of the lessons are still applicable to the ext2 filesystem covered on this exam. You can use the same commands

to format, size, and check ext2 and ext3 filesystems; the only difference is the addition of a journal. And the FHS has not changed significantly; Chapter 7 can help you here.

`cron` jobs are easy to schedule through scripts in the `/etc/cron.d` directory. You can make backups to tapes, CDs, other hard disks, and more. RAID is another way to protect your data. You need to be aware of several shared libraries, such as the latest GNU C compiler. Many other programs depend on shared libraries. System tuning in some ways is an art form; the optimal block size for your partitions depends on file sizes and how they are used. These techniques are covered in a number of chapters.

If you ever have a problem that keeps you from booting Linux, you have an emergency on your hands. It's important to know how Linux boots, from the MBR, the bootloader, through the kernel, from initial RAM disk, and more. The Red Hat–based rescue process is covered in Chapter 11; while other distributions may have dedicated rescue floppies, the principles remain the same. Once you've booted into a damaged system, you need to know how to repair filesystems and restore from backups.

Base Systems

For this section, you need to know how to manage users, groups, quotas, and file permissions and ownership. Red Hat's User Private Group scheme is not standard for Linux; you need to know how other distributions group users. For this purpose, it's useful to install a different version of Linux, ideally on another computer. You need to know how to manage ownership; the `chown` and `chgrp` commands can help at the user and group levels. User and group management techniques are covered in Chapter 9; quotas, permissions, and ownership are addressed in Chapter 6.

Most computer users want e-mail. You'll need to configure a POP or IMAP server; you can then configure mail clients such as `pine` or Evolution. Sometimes this involves aliases as people move or change jobs. These issues are explained in Chapter 21.

When Linux boots, it mounts filesystems based on `/etc/fstab`, finds a default runlevel in `/etc/inittab`, and then starts services in the `/etc/rc.d/rc`*`runlevel`*`.d` directory. Some distributions allow directories to be mounted on multiple partitions using Logical Volume Management. The runlevel determines which other services start during the boot process. Conversely, `shutdown`, `halt`, and `reboot` stop Linux by moving to runlevels 0 or 6. These processes are described in Chapter 11. You'll also need to know how to recompile the kernel as described in Chapter 12.

Shells and Commands

Other Linux distributions do not include Red Hat's safeguards on the root account; using root is discouraged for all but essential uses, which can be managed through `su` and `sudo`. You can then access the accounts of your choice and revise passwords.

There are a number of ways to communicate effectively with users; some involve messages sent during the login process, in `/etc/issue` and `/etc/motd`.

Some elements help Microsoft Windows users make the transition to Linux. The look and feel of GNOME and KDE provides a relatively user-friendly GUI. File managers such as Midnight Commander resemble Windows Explorer. The `mtools` allow you to use a number of DOS commands on vfat-formatted directories.

You should be able to recognize basic commands within a shell script. Many shell scripts are based on a series of regular shell commands, which are described in several chapters.Others can be compiled from source code with variations of the `make` command. Some include programming

commands with conditional statements that either loop or select an option. Programming is beyond the scope of this book.

The log files in `/var/log` are a rich source of information on your system; many log files are dedicated to specific services from Apache to Samba to the X Window. You also need to know a substantial number of command-line commands for this exam.

System Services

One group of system services relates to different types of packages. Programs come in packages such as RPMs, Debian DPKGs, and tarballs. Backups can be saved in tarballs and other package formats. Backup strategies keep you from having to back up your entire computer every time.

Another group of commands allows you to check running processes in different ways. Variations on `ps` can show you what is running with associated IDs. Use the `top` command to highlight processes that are loading your system. You can prioritize processes with `nice` and `renice`, and stop processes with various `kill` commands. These commands and files are described in Chapter 13.

For this exam, expect to know how to configure `/etc/printcap` in detail. This is a difficult file with an obscure language. It's part of the LPD system, and Red Hat is moving away from it. Therefore, it's not covered in significant detail in this book; however, it is well covered in the Printing-HOWTO of the LDP.

Applications

You need to know how to configure a number of Linux services. In this section, they relate to backups, display tools, e-mail, web services, window managers, FTP servers, SSH, newsgroups, and GUI tools associated with specific Linux distributions.

While you don't have to be a guru on any backup application for this exam, you should know about the existence of third-party tools such as Amanda, KBackup, UNiBACK, Taper, and Arkeia. These tools are over and above the basic tools described in Chapter 14.

Some display applications include display managers. Applications such as `xdm`, `gdm`, and `kdm` provide a login interface for a Linux GUI. Display managers are covered in Chapter 29. The VNC (Virtual Network Computing) system allows you to connect to the GUI of a remote Linux or Windows computer.

The exam focuses on sendmail, the most popular outgoing e-mail server. However, you should be aware of the alternatives, including Smail, Postfix, and Exim. There are incoming e-mail server alternatives to the POP3 and IMAP services in Chapter 21, such as QPopper and Mahogany.

The exam focuses on various components of the Apache web server, including the SSL, Perl, PHP server scripts, and frontpage modules. While these components are covered in Chapter 25, this exam was developed when Apache was at version 1.3. Red Hat Enterprise Linux 3 includes Apache version 2.0, so what you see on the exam may be somewhat different.

As discussed in Chapter 29, several window manager options are available. They provide the look and feel of a GUI on a desktop environment.

As described in Chapter 22, Red Hat supports vsFTP. This exam also requires basic knowledge of the WU-FTP, Pro-FTP and glFtpD FTP servers.

Troubleshooting

Most troubleshooting techniques apply to either the Installation and Configuration or the Networking exam. But as a system administrator, you can use commands such as `service` to check the status of many daemons. You can use `fsck` to inspect and fix filesystems. Log files in `/var/log` can help.

In Red Hat Enterprise Linux, when you're in GNOME or KDE, CDs are automatically mounted by default. If you can't even use the appropriate `mount` command for a CD, the first thing to check is `/etc/fstab`, per Chapter 7.

System resources can be limited by user based on the bash `ulimit` command. The corresponding Korn shell command is `rlimit`. This command is now obsolete in Red Hat Enterprise Linux. You can learn more about the Korn shell in *Learning the Korn Shell*, second edition, by Bill Rosenblatt.

Networking

The networking exam is fairly comprehensive; it's the only major Linux certification exam that is dedicated to this topic. The topics on this exam cover everything from physical networking to details of specific network services.

Theory of Operation

While this section addresses the theory behind networking, the concepts addressed are broad and wide. You need to know the basics of networking, from the way networks are built physically to the concepts behind the TCP/IP protocol stack. This includes a basic knowledge of IPv4 and IPv6 addresses and hardware addresses.

As you work your way through the TCP/IP protocol stack, this section covers some of the major services in detail, including DNS, NFS, UUCP, and Samba. You should know some basics of the alternative Novell IPX/SPX protocol stack. Many of these concepts are covered in Chapters 15, 16, 22, and 24.

Several other concepts go beyond Linux into the basic principles of networking. For more information on basic network concepts such as Internet topologies and bandwidth management, refer to a general text on networking.

Base Systems

This section can be divided into two areas. First, there are the basic physical network concepts, such as how networks are physically organized (topology), and network hardware. Second, there are the Linux interfaces and commands used for hardware and network addresses; firewalls and proxy servers; multicasts; tunneling and IP aliases. Many of these concepts are covered in Chapters 15–17 and other network-related chapters.

Most networks are organized in a "star" topology, where several computers are connected to a hub. Alternatives include bus and ring topologies. Other major network components include modems, network cards, hubs, switches, routers, and gateways.There is a focus on network cards of all types, from ISDN to common high-speed Internet (cable, DSL) to T1 connections

Computers talk to each other based on their hardware, or MAC addresses. These addresses are collected in ARP tables, which you can modify with the appropriate `arp` commands. The `ifconfig` and `route` commands let you assign new hardware addresses to network cards and determine the route to

find various IP addresses. With multihoming, you can use `ifconfig` to assign more than one IP address to each network card.

These routes are sometimes organized into routing tables. This allows you to configure your Linux computer as a router between different networks. Routers commonly exchange information through the Routing Information Protocol (RIP).

Linux computers that are configured as routers can also serve as firewalls that protect your LAN. Basic firewall scripts are based on the `ipchains` and `iptables` commands. The `iptables` system is also known as *netfilter*. The right `iptables` commands can stop the "ping of death," also known as *SYN flooding*. Other firewalls can be organized through TCP Wrappers and the Squid proxy server.

TCP/IP organizes services on different ports. Several services, such as NFS and NIS, need a portmapper to help. When you specify the right ports, you can use certain `iptables` commands to help you "tunnel" securely through a firewall. This is also the principle behind virtual private networking.

Just remember, the SAIR Networking exam was originally developed before `iptables` was in common use.

Shells and Commands

Unix was developed concurrently with the ARPAnet, which eventually became the Internet. TCP/IP was developed for Unix. Since Linux is a clone of Unix, it is well suited to the Internet. Linux includes a substantial number of network commands and services, described in Table 27.2. Chapter references are included if you need more information.

Table 27.2: SAIR Networking Shells and Commands

Command/Service	Function	Chapter
`arp`	A command that associates hardware and IP addresses.	16
`finger`	User information from `/etc/passwd`.	18
`ipchains`	Firewall (such as `iptables`).	17
`ftpd`	The FTP server daemon; see vsFTP or WU-FTP.	22
`httpd`	The Apache server.	25
`ifconfig`	Network card configuration.	16
`inetd`	The obsolete version of `xinetd`.	18
IPX	Part of the IPX/SPX protocol stack.	15
`logd`	An older logging daemon; Red Hat uses `syslogd`.	13
`lpd`	An older print daemon; Red Hat's default is CUPS.	n/a
`mail`	E-mail client; `mutt` is an alternative at the command line.	21
NIS	The Network Information Service database.	23
`named`	The DNS server daemon.	19

Continued on next page

Table 27.2: SAIR Networking Shells and Commands *(continued)*

Command/Service	Function	Chapter
`netstat`	A command that tells you the network status.	16
NFS	Network File System.	22
`nslookup`	A command that performs DNS database lookups; replaced by `dig`.	19
`ping`	A command for checking connectivity.	16
`pppd`	The Point-to-Point Protocol daemon, primarily for telephone modems.	16
`dhclient`	The DHCP client; successor to pump and `dhcpcd`.	19
`rsh`	Remote shell commands.	
sendmail	The outgoing e-mail server.	21
Samba	A service for sharing with Microsoft computers.	24
SSH	Secure Shell.	18
`tcpdump`	The command associated with Ethereal.	17
Telnet	Remote connections.	18
`traceroute`	A command for checking the integrity of a network route.	16
`uucp`	Unix-to-Unix copy for outgoing e-mail.	n/a
`aliases`	Also known as virtual e-mail users for sendmail.	21

System Services

This section of the SAIR Networking exam requires you to know a number of system services in some detail. Be prepared to understand the workings behind the associated configuration files. The file locations noted in this book are good for Red Hat Enterprise Linux 3; they may be elsewhere on a different Linux distribution (or even an older version of Red Hat Enterprise Linux). These services are listed in Table 27.3. Chapter references are included if you need more information.

Table 27.3: SAIR Networking System Services

Service	Function/Configuration File	Chapter
DNS	Name resolution: `/etc/named`, `/etc/resolv.conf`, `/var/named`	24
FTP	File transfers; protected by `/etc/ftpaccess`, `xinetd`	27
NFS	Linux/Unix file sharing; `/etc/exports`	28
`xinetd`		23
Samba	Windows file sharing; `/etc/samba/smb.conf`	29

Continued on next page

Table 27.3: SAIR Networking System Services *(continued)*

Service	Function/Configuration File	Chapter
Sendmail	Outgoing e-mail; `/etc/mail/sendmail.cf`	26
POP3, IMAP	Incoming e-mail; protected by `xinetd`	26
Mailing List Servers	ListProc, Majordomo	
Apache	Web service; `/etc/httpd/httpd.conf`	30

Applications

In Linux networking, applications are the programs you use to connect to services. The SAIR Networking exam tests your knowledge of these applications for e-mail, the X Window, browsers, Samba connections, FTP, and network configuration tools.

The SAIR Networking exam assumes that you can open and close command-line mail and news clients such as `pine` and `trn`. It tests your knowledge of X Window management with the X server on the remote computer protected through `xhosts`, as described in Chapter 29.

While Netscape is no longer part of the Red Hat Enterprise Linux 3 distribution, you can still install it on the Linux operating system. Other available browsers such as Mozilla and Konqueror are briefly covered in Chapters 30.

Samba includes an administration tool, SWAT, which is itself a web browser–based application, functionally similar to `redhat-config-samba`. Two related applications, `smbclient` and `smbmount`, connect you to a Samba server. You can learn more about these Samba tools in Chapter 24.

As with Samba, FTP clients are applications. Two examples are the `ftp` command, which you can use at the command-line interface, and gFTP, which you can use from a GUI. You can learn more about these FTP clients in Chapter 22.

Troubleshooting

Network troubleshooting issues on the SAIR Networking exam are nearly as diverse as on the Installation and Configuration exam. Many of these items are covered in different chapters in this book. Some of these issues include the following:

- You should know how to install network card drivers with `insmod` and `modprobe`.
- Be able to diagnose `/etc/printcap` for remote printers; remember, this is related to LPD, not the more current CUPS print service.
- Commands such as `dig` can help you diagnose DNS database problems.
- Sometimes computers on your LAN are not reachable due to bad IP or hardware addresses.
- Packets are often dropped at various locations between computers; `netstat` and `traceroute` help you isolate the problem.
- FTP downloads can be stopped via `/etc/ftpaccess`.

- Distant network connections aren't always reliable; `traceroute` helps you identify the LAN/router that's losing your message.
- Some network cards send out data constantly, a.k.a. *chattering*; this can stop communication from other computers on your LAN.
- You can configure sendmail to verify destination e-mail servers; problems can result in "relaying denied" messages.
- Network cards aren't always detected properly; the first place to check is in the `dmesg` output.
- If you have a shared NIS database and can't log into some clients, you may have a `ypbind` or `ypserv` problem.
- Because of collisions, the actual data transmission speed is usually much slower than the maximum on a busier Ethernet network.
- User e-mail can be forwarded via `/etc/mail/aliases`.
- If sendmail can't verify destination servers, it may hang your system. This issue is similar to what happens when an NFS client can't find a mounted directory from an NFS server.
- NFS should not block the local `/root` directory; however, remote root user access is normally mapped to the nobody user.
- Sometimes you need to update the `/etc/hosts` or DNS database.

Security, Ethics, and Privacy

SAIR's final LCA exam is known as Security, Ethics, and Privacy. If you're afraid of the risks associated with the Internet, one simple solution is to never connect your computer to your ISP. You don't even need to connect your computer to any others on a LAN.

Yet computer networks are a fact of life today. Many of the things you do to secure a network bring up ethical questions. For example, ethical Linux administrators who find someone's password with Ethereal won't use that password themselves but will advise that person to use a more secure service.

Most of the security exam is related to the ways you encrypt and secure data on a LAN. Some test your knowledge of tools that help you test the security of your network.

Theory of Operation

The basics of computer and network security are covered in Chapter 17. Some best practices are beyond Linux; for example, it makes sense to keep some systems and hardware in physically secure locations. Most networks connected to the Internet will be attacked. Programs are available that search the Internet automatically looking for vulnerable systems.

There are things you can do to minimize the risks to your LAN. Encryption, using techniques such as the GNU Privacy Guard (GPG), allows you to scramble messages between computers. Strong passwords can take weeks to crack; frequent changes can make them even more secure.

You can secure your files in different ways. Some distributions support Access Control Lists for files and directories. The right `umask` value minimizes rights for unauthorized users. A properly configured

firewall can filter unwanted data types and help protect your system from excessive data flow, such as the ping of death.

As an administrator, you should monitor your system regularly for suspicious activity. Some monitoring is possible through log files; many administrators use third-party tools to check system security. Keep track of the latest information from the Computer Emergency Response Team and Coordination Center (CERT/CC) at `www.cert.org`. And follow best practices with respect to passwords and physical security, within the limits allowed by your own national laws.

Base Systems

This section defines the basic systems you can install and use to help secure your system. These systems and services include the following:

- The Shadow Password Suite encrypts user and group passwords in files readable only by root. It is installed by default on Red Hat Enterprise Linux 3. You can learn more about the Shadow Password Suite in Chapter 9.
- System log files help you detect suspicious activity. Different Linux distributions may organize them differently in `/etc/syslog.conf`, and they may maintain them differently through `/etc/cron.d`. The standard Red Hat log file configuration is described in Chapter 13.
- TCP Wrappers and `iptables` firewalls can block traffic to TCP/IP services and or ports. These firewalls are discussed in Chapters 17 and 18.
- Vulnerabilities are often addressed through patches on the kernel. Patching techniques are discussed in Chapter 12.

Shells and Commands

Linux includes several commands related to security. The topics can be grouped into the following areas: administering users; monitoring logs, communication, and services; maintaining files; and encrypting data.

Even if you trust your users, people do make mistakes. Crackers who take over a user account can cause trouble. They may assign a UID of 0 to a regular user account, which gives it root privileges. Commands such as `who` and `w` help you monitor current users; `lastcomm`, which is part of the Red Hat `psacct-*` RPM, allows you to monitor commands by terminal. Some users can get partial or full root privileges with the right settings in `/etc/sudo`. Needless to say, this involves some risk.

As discussed in Chapter 13, it's useful to monitor certain log files in `/var/log`; if you suspect a problem, you can focus on the log data that you collect through `/etc/syslog.conf`. You can also monitor communications, as described in Chapter 17, with tools such as Ethereal. Many services have their own log files; others you can monitor with the appropriate `netstat` command.

Crackers sometimes substitute viruses, worms, and/or Trojan horses for critical files. Chapter 10 describes RPM commands that compare current files against the originals in the package. Perhaps key to this are files associated with authentication, such as `/etc/passwd`.

Encryption keeps users from reading data as it travels over a network. The SSH commands allow you to log into remote computers securely through an encrypted connection. You can learn more about SSH in Chapter 18.

System Services

A couple of other systems help secure your computer and LAN, and several third-party services are available that help you identify flaws in your security. In principle, this starts by disabling or uninstalling services that you do not use. These systems and services include:

- Firewalls associated with `ipchains` (`iptables` were not yet released when the SAIR exams were developed) support direct security. Such firewalls can also be used to filter unwanted data.
- TCP Wrappers support security on a per-daemon basis.
- UUCP is associated with different mail protocols.
- Other services can be secured through appropriate configuration files, including Apache, PPP, Telnet, FTP, and SMTP.
- Other systems can be secured, including the POP3 and IMAP4 mail services, as well as NIS.
- Kerberos is a system that uses "tickets" to allow computers to exchange data securely through a public network. It requires a ticket-granting server and an appropriate client.

Applications

You can configure your applications and services to make them secure. Third-party tools are available that help you measure the security level of your services. This section of the exam criteria also examines Linux distributions that are reportedly more secure than the average distribution.

You can secure parts of a website on Apache; some areas or even virtual sites can be protected with usernames and passwords. With the proper configuration, if crackers do manage to break into your Apache files, they'll be trapped in that directory by a chroot jail. These provisions are discussed in Chapter 25 (We describe chroot in Chapter 11).

There are a number of Linux distributions that are reportedly more secure; the most well known is NSA Linuxkernel and packages, developed by the U.S. National Security Agency. Other "secure" distributions include Bastille and Trustix. More information on these is available through their websites, shown in Appendix A. In addition there are tools which can help you counteract the effects of trojan horses, worms, and viruses:

- Third-party tools, such as Crack and Tripwire, help you check the integrity of your passwords and files, respectively.
- Other third-party tools, such as PortSentry and Sysmon, let you monitor a system for security surprises.
- File encryption, when used with md5 checksums, can help you verify the integrity of files and directories.

Troubleshooting

Troubleshooting security means having a backup plan. In other words, what do you do if security fails? If all else fails, a reliable backup allows you to restore your system to a state before the security breach. But backups are not enough. Daily cron jobs, when you check the results, can help you identify security problems.

You should use all the tools at your disposal to try to identify the cracker and/or his computer. If you can identify the source, you can add explicit commands in your firewalls to block future access by that cracker. Remote denial of service attacks can be blocked with a straightforward addition to your firewall.

When you have restored your system, check it with the appropriate file verification tools. A backup does you no good if it includes Trojan horses or worms masquerading as key files.

Make sure that users follow appropriate security procedures. This is a balancing act; if security is too tough, users may become motivated to find a way around your security rules.

And make sure that you follow appropriate security procedures. This includes making sure you have appropriate updates to your Linux kernel and other applications.

Summary

There are three major generic Linux certification programs: CompTIA's Linux+, LPI's Level I certification, and SAIR's Linux Certified Administrator. The Linux+ exam is generally recognized as more suitable for newer Linux users.

The Linux+ exam is currently undergoing major revisions. As of this writing, the plan is to include topics from six domains: Installation, Management and Maintenance, Configuration, Security, Documentation, and Basic Linux Hardware. As this exam is "distribution neutral," it's focused on the command-line interface. As the revisions are targeted at users with 6–12 months of experience, expect higher-level commands than you may have seen before. The Linux+ exam will still include a significant (but reduced) group of questions on PC hardware that goes beyond Linux; many of these questions are similar to those you may find on CompTIA's A+ Hardware exam.

LPI requires you to pass two exams for its LPIC-I certification. These exams are known as General Linux I and General Linux II. LPI believes that candidates who pass their exams are qualified to be junior Linux system administrators. The LPI exams are heavy on the command line; objectives require you to know how to configure every major service, though not necessarily in any advanced detail. The LPI exams were recently revised, so the questions are essentially up-to-date with the latest Linux distributions. The emerging Novell certifications are in part based on the LPI exams.

SAIR requires you to pass four exams for its LCA certification. SAIR also believes that candidates who pass their exams are qualified to be junior Linux system administrators. These exams are: Installation and Configuration; System Administration; Networking; and Security, Ethics, and Privacy. The objectives for each of these exams are divided into the same six areas: Theory of Operation, Base Systems, Shells and Commands, System Services, Applications, and Troubleshooting.

In Chapter 28, we'll look at the Red Hat Enterprise Linux certifications, the Red Hat Certified Engineer (RHCE) and Red Hat Certified Technician (RHCT). They are fairly unique exams; they require candidates to demonstrate real-world debugging and configuration skills at an actual Linux computer.

Chapter 28

Red Hat Certifications

THERE ARE NOW THREE Red Hat certification exams The Red Hat Certified Engineer (RHCE) exam is fairly well known as an elite "hands-on" exam. At the beginning of 2003, Red Hat also released an exam for newer Linux administrators, the Red Hat Certified Technician (RHCT) exam. As of this writing, Red Hat announced the upcoming release of the Red Hat Certified Architect (RHCA) exam, slated for release in early 2005.

The Red Hat exams test more than just knowledge. They test your competence as a Linux administrator. During the exam, you're put in front of a computer with realistic problems that you have to debug. You are asked to install Red Hat Enterprise Linux and a number of services, with challenging specifications.

All three Red Hat exams are for experienced Linux users. They are extremely difficult for users who are new to Linux. Red Hat includes a set of broad prerequisites, which seem designed to weed out those newer Linux users. But you don't have to know everything. Many seasoned Linux administrators aren't experienced in all areas. With a little extra study, they can still succeed on the Red Hat exams.

This chapter provides a general overview of the Red Hat exams and is not intended as a substitute for a test preparation book or Red Hat's RH133 (RHCT), or RH300 (RHCE) test preparation courses, or the multiple courses associated with the RHCA. exam preparation courses. The course number for the RHCA is TBD as of this writing; several courses may in fact be required for this most advanced Red Hat certification. This chapter focuses on the Red Hat exams, not Linux itself. For more information, chapter references are provided.

The RHCT exam is a complete subset of the RHCE exam. It is also a "hands-on" exam directed toward administrators who are installing Linux on desktops and workstations. It does not require detailed knowledge of configuring Linux as a network server.

However, Red Hat has stated that while the RHCE will be prerequisite to the RHCA, the RHCE exam requirements are not a direct subset of the RHCA. If you're interested in this certification, monitor Red Hat's website for details in late 2004 or early 2005. We summarize these exams in Table 28.1.

Table 28.1: Red Hat Certification Exams

Certification	Description
RHCT	The Red Hat Certified Technician, which includes Installation and Configuration, as well as Troubleshooting and System Maintenance exams.
RHCE	The Red Hat Certified Engineer, which includes Installation and Configuration, as well as Troubleshooting and System Maintenance exams. Includes all RHCT requirements.
RHCA	The Red Hat Certified Architect; exam details to be announced in late 2004 / early 2005.

I have written books on several Linux certification programs and believe that you can use this book to help supplement your studies. This chapter covers the following topics:

- Looking over the Red Hat Exams
- Exploring the prerequisites
- Understanding the RHCT exam
- Preparing for the RHCE exam

Looking Over the Red Hat Exams

The Red Hat exams are mentally demanding challenges. For both the RHCT and RHCE exams, you have to solve problems on an actual computer, and install and configure Red Hat Enterprise Linux.

Both exams include two parts: Troubleshooting and System Maintenance exam and the Installation and Configuration exam. As both parts are serious challenges, I also refer to these as exams.

An Overview of the RHCT Exam

The RHCT exam tests your ability to install, configure, and attach a Red Hat Enterprise Linux 3 computer to an existing production network. This process is associated with a workstation installation.This exam includes two parts:

- Troubleshooting and System Maintenance (1 hour)
- Installation and Configuration (2 hours)

To pass this exam, you need to solve all five problems on the Troubleshooting and System Maintenance exam. You also need to install and configure enough Linux components to get a score of at least 70 percent on the Installation and Configuration exam.

An Overview of the RHCE Exam

The RHCE exam tests your ability to install and configure Red Hat Enterprise Linux, work within any limitations of your hardware, configure filesystems in server configurations, set up and configure networking and network services, and demonstrate basic administration skills. It also tests your ability to maintain and troubleshoot these configurations. This exam includes all the requirements of the RHCT exam and consists of two parts:

- Troubleshooting and System Maintenance (2.5 hours)
- Installation and Configuration (3 hours)

To pass the RHCE Troubleshooting and System Maintenance exam, you need to solve *all* RHCT-level problems within the first hour on the Troubleshooting and System Maintenance exam. You also need to solve enough RHCE-level problems for an overall grade of 80 percent on this exam.

You also need to install and configure enough Linux components to get a score of at least 70 percent on *both* the RHCT and RHCE portions of the Installation and Configuration exam.

On the Troubleshooting and System Maintenance exam, you're actually put in front of a broken Linux computer. You'll either get or will need to create an appropriate boot or rescue disk, and then you must diagnose and fix any problems.

On the Installation and Configuration exam, you are told to install Red Hat Enterprise Linux. That's easy enough. But you are also told to install and configure a substantial number of services. This is quite a challenge for many to complete in two or three hours (depending on the exam). However, there are different ways to configure Linux; as long as you meet the specifications set out in the exam question, you'll get credit for what you do. The basic Red Hat Exam Prep Guide is available online, at `www.redhat.com/training/rhce/examprep.html`.

The RHCE exams are "closed book." You're not allowed to bring any notes with you into the exam room. However, on the Troubleshooting and Installation exams, you are allowed to refer to any man pages and documents that you may find on the computer where you're installing Red Hat Enterprise Linux during the exam.

NOTE *If you don't pass the RHCE exam, you may still score well enough to get the RHCT credential. On the Troubleshooting and System Maintenance exam, you need to solve all of the RHCT problems. Also, you need a score of at least 70 percent on the RHCT skills on the Installation and Configuration exam.*

Exploring the Prerequisites

Red Hat has a list of prerequisites for people who want to become Red Hat certified. While these topics are quite broad, they should not intimidate you. Many experienced Linux system administrators don't know every prerequisite topic in depth. With a period of self-study, you can fill in any gaps in your knowledge.

The prerequisites are what Red Hat believes you should know—before taking one of their exam preparation courses. If you're not comfortable with many of the topics in the list, read the other chapters in this book. Alternatively, you can take one or more of the Red Hat introductory courses on Linux.

While I've taken most of this section from the official prerequisites for the RHCE course, RH300, you can see from the Red Hat Exam Prep Guide that these prerequisites apply to *both* exams.

As you'll see in the following sections, the prerequisites before you work with any of the Red Hat exam curricula run the gamut from basic PC hardware through system administration, network services, and security.

Red Hat Skills Courses

Red Hat offers several courses for people who want to prepare for the RHCT or RHCE exams. These courses include:

RH033, Red Hat Linux Essentials, is for people with no experience working at the Linux or Unix command-line interface and want to use and customize a Linux workstation.

RH131/RH133, Red Hat Linux System Administration, is for users who want to build their skills to where they can connect and configure a Linux workstation on an existing network. The RHCT exam is included in RH133.

RH202, RHCT Exam, is the course you register for when you want to take just the RHCT exam without instruction.

RH253, Red Hat Linux Networking and Security Administration, is for users who want to build their skills at configuring common network and security services.

RH300/RH301, RHCE Rapid Track Course, is for users who need a refresher before taking the RHCE exam. The exam is included in RH300.

RHS333, Red Hat Enterprise Security: Network Services, goes beyond the security features covered for the RHCE, and is part of the preliminary curriculum for the RHCA, as of this writing.

RH401, Red Hat Enterprise Deployment and Systems Management, is for architects who deploy and manage Red Hat systems in the enterprise, and is part of the preliminary curriculum for the RHCA, as of this writing.

RH423, Red Hat Enterprise Directory Services and Authentication, provides LDAP and PAM skills associated with authentication and authorization, and is part of the preliminary curriculum for the RHCA, as of this writing.

RH436, Red Hat Enterprise Storage Management, is designed to provide experience with configuring Red Hat systems in clusters, using Shared Storage technology. It is part of the preliminary curriculum for the RHCA, as of this writing.

RH442, Red Hat Enterprise System Monitoring and Performance Tuning, is designed to teach the methodology of performance tuning and capacity planning for Red Hat Enterprise Linux. It is part of the preliminary curriculum for the RHCA, as of this writing.

This is not a full list of the offerings by Red Hat. These courses are offered at an expanding number of locations worldwide; for more information, navigate to the Training section of `www.redhat.com`.

Even though Red Hat no longer produces a "Red Hat Linux" distribution, the courses still have this title. All these courses are now taught with Red Hat Enterprise Linux.

Basic Hardware Knowledge

Every computer administrator needs some basic knowledge of PC hardware. For example, you should know the standard channels used to communicate on a PC. And before you start organizing partitions, be familiar with how IDE and SCSI hard drives interact with your PC. These topics are covered in Chapter 2.

PC hardware starts with the type of CPU. While most PCs work well with the standard i386 Linux kernel, customized kernels are available for many types of CPUs.

Hardware Communication

PC components communicate with each other in three basic ways: IRQ ports, I/O addresses, and DMA channels. Some IRQ ports are always assigned to key components, such as the system clock. Others are available for less essential parts of the PC. The same is true for I/O addresses. Configuring Linux for your PC is often an exercise in managing these ports, addresses, and channels.

Hard Drives

Two basic types of hard drives are in common use today: IDE and SCSI. The IDE hard drive (also known as ATA) is the standard that comes with most regular PCs today. Unfortunately, PCs are limited to four IDE drives. SCSI drives are generally faster and more flexible; you can install up to 32 SCSI hard drives on your PC. IEEE 1394 hard drives are a variation on the SCSI standard.

Before installing Linux, you must assign a primary hard drive. That's where you'll install a bootloader such as GRUB. Next, you can plan how you're going to organize partitions. If you're going to install RAID or Logical Volume Management (LVM), you can even assign a filesystem such as `/home` to partitions on multiple physical hard drives.

Basic Linux/Unix Knowledge

Red Hat focuses on the `vi` text editor. It may be the only editor available if you ever have to rescue your system using a boot floppy. You may have noted in Chapter 27 that the other Linux certification programs also focus on `vi` to the exclusion of more popular text editors.

If you don't know `vi`, learn it, at least to the level described in Chapter 6. When you're editing a configuration file on either of the Troubleshooting and System Maintenance exams, `vi` could be your only choice for fixing any problems that arise.

Filesystem Hierarchy

Linux directories are organized into the Filesystem Hierarchy Standard (FHS). When you divide your SCSI and IDE hard drives into partitions, each partition gets a specific `/dev` file. CDs and DVDs get their own `/dev` files. You can assign different FHS directories to each of these devices.

Other storage media also get their own `/dev` files. When you configure a directory on a partition, the associated device gets a label. You can assign and inspect this label by using the `e2label` command. Partitions can be mounted or unmounted based on related specifications in `/etc/fstab`.

Partitions are organized with Disk Druid during the installation process, or `fdisk` at any time. They are formatted with `mkfs` and checked with `fsck`. With LVM, you can even configure a filesystem such as `/` on multiple physical drives. More information on these topics is available in Chapters 2, 3, and 7.

Basic File Operations

Two types of basic Linux commands are described in the Red Hat prerequisites. One type allows you to navigate, to read and find files, and to manage basic packages. These commands include `cp`, `mv`, `ls`, `more`, `less`, `cd`, `find`, and `tar`.

The other type enables you to filter information. Commands such as `grep`, `wc`, `head`, and `tail` allow you to look through existing files and data for useful information. The `grep`, `sed`, and `awk` commands allow you to search and process text data in more detailed ways.

To make Linux commands effective, you need to understand Linux wildcard concepts. Key wildcards includes `*` and `?`; it's also possible to specify a group or range of different options in brackets, such as `[135]` or `[a-d]`. These concepts are also known as *globbing*.

These are the commands that you probably use every day as a Linux administrator. For more information, see Chapter 6.

Printing

Red Hat has recently changed the default print service from the Line Print Daemon (LPD) to the Common Unix Print System (CUPS). While CUPS is associated with a web browser–based interface, it uses basic commands similar to LPD at the console. For example, while LPD is associated with the `lpq`, `lpr`, and `lprm` commands, CUPS is associated with `lp`, `lpr`, and `lpoptions`. In fact, the `cups-lpd xinetd` service allows older applications that use LPD commands to work with CUPS. Both groups of commands are discussed in Chapter 20.

You should also know how to add printers locally and remotely. The `redhat-config-printers` tool can help you set up LPD printers. Once the CUPS service is active, you can navigate to its configuration tool by directing your browser to `localhost:631`.

Understanding the Shell

If you're a Linux administrator, you work with the shell. You create your own scripts to automate tasks such as backups. As a skilled administrator, you combine commands. You know how to customize your shell environment to best meet your needs.

Red Hat Enterprise Linux includes a number of scripts in `/etc/cron.*` directories that are run automatically, per `/etc/crontab`. The structure of the `cron` daemon described in Chapter 13 can help you organize the scripts with appropriate permissions required to administer a Linux network.

Commands can be combined; data can be taken from or sent to various files. The processes of piping, standard input, standard output, and standard error are described in Chapter 6.

The shell environment includes defaults when you log into Linux and other variables and parameters that you can set. The Red Hat exams assume you know how to find these defaults for the bash shell. Perhaps the most important parameter is the PATH, which determines where Linux searches for commands in your system.

Security

The security prerequisites on the RHCE exam include four basic concepts. These concepts are discussed in more detail in Chapters 6 and 9.

- The Shadow Password Suite hides user and group passwords in files readable only by the root user.

- Every file includes a set of permissions for the owner and the group that owns the file. There are also permissions for other users on your system.
- To understand permissions, you need to understand how users and groups are organized. The files that users create are affected by the applied value of `umask`.
- Permissions can be modified; the SUID and SGID bits are commonly set when you want to share access to a program or a directory.

System Administration

If you're reading this book, you probably want to learn more about administering Linux. This book and the Red Hat exams cover all sorts of system administration skills. This section simply includes those topics that are difficult to classify in other areas.

Red Hat Enterprise Linux allows you to configure a common set of files for all new users in `/etc/skel`. You can add the files and directories that you or your organization may want everyone to have. For more information on how this works, see Chapter 9.

Daemons are processes that usually run in the background. For example, Apache starts a number of daemons; more are started when more users try to connect to your website. Daemons are generally organized in the `/etc/rc.d` directory and managed with tools such as `chkconfig`.

Perhaps the key administrative daemon is `cron`, which can help you schedule jobs to be run at any time, day or night. This is controlled by `/etc/crontab`; alternatively, users can configure and control their own `cron` jobs with the `crontab -e` command. These jobs are stored in user files in `/var/spool/cron`.

Linux logs are stored in `/var/log`, based mostly on `/etc/syslog.conf`. Log files are normally maintained by the `logrotate` `cron` daemon. Logging, daemons, and `cron` are discussed in Chapter 13.

With most Linux distributions, even administrators run most commands as regular users. The superuser concept allows you to run a limited number of commands as a root user, limiting your risks. You can assume root privileges with the `su` command (and the root password), or users can obtain limited root privileges with `sudo`, as configured in `/etc/sudoers`.

Linux administrators are often responsible for protecting the data on the computers on a LAN. One way to do this is with backups. Chapter 14 describes various backup methods. Command tools that can help with this process include `tar`, `gzip`, and `bzip2v`.

Basic Networking

The basic protocol stack for Linux is TCP/IP. When you configure a Linux computer for networking, you must know three basic things: Every computer on a TCP/IP network gets its own IP address. You can configure and test the connection with several different commands. The configuration is documented in a series of files in the `/etc` directory. To learn more about IP addressing, the associated commands, and most of the configuration files, see Chapter 16.

An IP address is not enough. Every computer on a TCP/IP network also needs a network address, a broadcast address, and a network mask. It often also needs a gateway address, and maybe even the IP addresses of DNS servers.

While IP version 6 (IPv6) addresses are coming into common use, IP version 4 (IPv4) addresses still work. The concepts of assigning IPv4 addresses are well known and work well on even very large private networks.

There are several key TCP/IP configuration files on a Red Hat Enterprise Linux computer, including `/etc/hosts`, `/etc/resolv.conf`, `/etc/host.conf`, and `/etc/nsswitch.conf`. Key configuration commands include `ping`, `ifconfig`, and `netstat`.

Network Services

Configuring network services is a key part of both Red Hat exams. Naturally, you'll be configuring clients for the RHCT exam, and adding servers on the RHCE exam. Before you're ready to prepare for your exam, you need to know some basics of configuring key Linux network services. Several chapters in this book address the topic of configuring these services. Most of these services are controlled by scripts in the `/etc/rc.d/init.d` directory. These services include:

- NFS allows you to share directories on a network with Linux and Unix computers; the key configuration file is `/etc/exports`.
- sendmail lets you set up a server for outgoing e-mail; the key configuration file is `/etc/mail/sendmail.cf`.
- POP and IMAP are incoming e-mail server services controlled through a script in the `/etc/xinted.d` directory.
- FTP allows you to share files with users on other computers. It can be configured for anonymous or user/password access. Several FTP servers are available; we cover Red Hat's vsFTP server in Chapter 22.
- DNS includes a database of hostnames and IP addresses (usually) on larger networks. DNS servers on the Internet can exchange and refer to each other for more information; the key configuration file is `/etc/named.conf`.
- DHCP allows you to regulate the use of IP addresses on a network; the key configuration file is `/etc/dhcpd.conf`.
- SMB can be configured to share directories in a mixed network of Linux and Microsoft Windows computers; the key configuration file is `/etc/samba/smb.conf`.
- Apache, also known as `httpd`, is the most popular web server on the Internet; the key configuration file is `/etc/httpd/conf/httpd.conf`.
- NIS allows you to share a common database of configuration files with other Linux and Unix computers; the key configuration file is `/etc/ypserv.conf`.
- The `xinetd` daemon controls a number of services based on configuration files in the `/etc/xinetd.d` directory; it's the successor to the so-called Internet Super Server (`inetd`).

Network Clients

There are several network clients included in the prerequisites cited in the Red Hat Exam Prep guide. Generally, these clients should be elementary if you have even a little experience administering Linux.

- E-mail clients such as Mozilla or Konqueror are as easy to configure as the Microsoft counterparts.

- Mozilla includes much of the same code as the Netscape web browser and is as easy to use.
- The `lftp` client described in the Exam Prep Guide is an extended version of the standard FTP client.

Basic Network Security

When you secure a network, you're blocking unwanted data. Generally, that means you create a firewall that blocks all data, opening channels only for the data you want. These channels normally correspond to the ports associated with the TCP/IP protocol stack. A list of assigned ports is available in `/etc/services`. he concepts of network security.

There are three basic ways to create a Linux firewall. The most common is at the kernel level with the `iptables` command. You can use `iptables` to specify data by port, by protocol, or by computer. Other services can be blocked through commands in `/etc/hosts.allow` and `/etc/hosts.deny`. Several services include their own configuration files, which may block access by user and by computer.

One more way to secure a LAN is to give it private IP addresses. As discussed in Chapter 16, several ranges of private IP addresses are available; using the right `iptables` masquerading commands, you can hide the addresses of the computers on your LAN from the ravages of the Internet.

Understanding the RHCT Exam

If you're planning to take the RHCE exam, read the following sections. The RHCE exam tests you on all RHCT requirements.

The Red Hat Certified Technician (RHCT) exam explicitly covers only a portion of the requirements of the RHCE exam. The following sections briefly describe what the RHCT exam does and does not cover. The technical details were described earlier in this chapter.

The prep course for the RHCT exam is Red Hat course RH133, Red Hat Enterprise Linux System Administration. As of this writing, RH133 is a four-and-a-half-day course. The last day includes three hours for the RHCT exam.

As of this writing, the Red Hat website states that the RHCT exam is "a realistic performance-based lab exam" that tests the individual's "ability to install, configure, and attach a new Red Hat Enterprise Linux system to an existing production network."

The RHCT exam consists of two parts. It begins with the one-hour Troubleshooting and System Maintenance exam with five problems. It ends with the two-hour Installation and Configuration exam.

The RHCT Troubleshooting and System Maintenance Exam

On the RHCT Troubleshooting and System Maintenance exam, some training companies suggest you'll need to solve all five problems. Red Hat requires that you solve all five problems to pass this part of the RHCT exam. If you're taking the RHCE exam, you'll also have to solve five RHCT problems on your Troubleshooting and System Maintenance exam.

According to the Red Hat Exam Prep guide, the problems you see can be from any of the six categories in this section.

Troubleshooting on Red Hat Exams

It's likely that this is the part of the Red Hat exams that promotes the most fear. If you don't have much in the way of hands-on experience, you probably haven't seen a lot of what can go wrong on Linux. In fact, Linux is so reliable that many administrators are not comfortable with the Troubleshooting and System Maintenanceexam.

Judgment and time management skills are required. If you're taking too long on a problem, you may want to give up and move on to the next problem. However, you can't go back. You lose the chance to get any credit for what you've done. And you could just be moments away from solving the problem.

Booting Into Different Runlevels

You'll need to know how to boot into different runlevels for troubleshooting and system maintenance. Typically, that means you'll have to boot into runlevel 1, which you can do by using the `init 1` command or by rebooting into single-user mode as discussed in Chapter 11.

But as you'll be configuring a Linux workstation, it may be set to boot into the GUI, which is runlevel 5. If you have problems booting into the GUI, one approach is to restart your computer and boot into runlevel 3 through the bootloader menu. In that situation, you can diagnose problems with the GUI; you can try the configuration tools described in Chapter 29.

Correcting Misconfigured Networking

When you connect your computer to a network, the connection works only if your network configuration is in order. You can use basic networking tools such as `arp`, `netstat`, and `ifconfig`; or you can use GUI tools such as `redhat-config-network`. These tools can help you diagnose and repair misconfigured networking on your computer. In either case, you should confirm the results in the basic network configuration files, which you may need to edit directly. Many of these files and commands are described in Chapter 16.

Correcting Hostname Resolution Problems

Part of any network configuration is hostname resolution. First, you can configure your computer's hostname in `/etc/sysconfig/network`; that won't work unless your `/etc/hosts` and network configuration files are consistent. Many of these files and commands are described in Chapter 16.

Configuring the X Window and Desktop Environment

When you configure a Linux workstation, you may need to accommodate users who have no skills at the command-line interface. Therefore, you'll probably need to configure and possibly diagnose problems with the X Window and Desktop Environment. The basics are to configure the X Window using the Display tool, also known as `redhat-config-xfree86`. Alternatively, you can configure the X Server configuration file, `/etc/X11/XF86Config`, directly. We show you how this is done in Chapter 29.

Note *You may note that some of the latest Linux distributions, including Fedora Linux, no longer use the XFree86 server. As Fedora Linux is used as a test bed for future Red Hat changes, it's reasonable to assume that Red Hat Enterprise Linux 4 will no longer use the XFree86 server.*

For users, you may also want to configure the desktop environment, including menus, GUI applications, and desktop icons. You can refer to Chapter 30 for more information on configuring the GUI desktop on a Linux workstation.

Adding Partitions, Filesystems, and Swap Space

Users collect data. As a Linux workstation administrator, you may need to reconfigure the partitions on a workstation. You can then mount the filesystems of your choice on those new partitions. If you add more memory, you may also want to add swap space. You probably did some of this during the installation process with Disk Druid.

However, you won't have time to reinstall Linux, as you have an hour to solve all five problems. It's fastest to use commands such as `fdisk`, `mkfs`, `mkswap`, and `mount` to set up, format, and configure partitions on your computer.

Remember that you'll need to document the changes in `/etc/fstab` to make the changes permanent; otherwise, you may not get credit for your work. You can learn more about these tools in Chapter 7.

Using Command-Line Tools

The fastest way to solve almost any problem on a Linux computer is at the command-line interface. When you have a problem on the Troubleshooting and System Maintenance exam, you'll have one hour to solve five problems. As stated in the RHCT part of the Exam Prep Guide, you'll need to use standard command line tools to analyze problems and configure your system. That could mean anything; however, if you know your commands, log files, and essential configuration files, you should be able to handle anything that Red Hat throws at you on the RHCT exam.

The RHCT Installation and Configuration Exam

During the RHCT Installation and Configuration exam, generally it's fastest to configure as much as possible during the installation process. As defined in the Red Hat Exam Prep Guide, you'll need to be ready to install over a network, as defined in Chapter 4.

Read your instructions carefully. You may need to create custom partitions as defined by your particular exam. Pay careful attention to RAID partitions. It's faster to configure RAID during installation instead of after installation as described in Chapter 14.

Note *Once Anaconda starts installing the hundreds of Red Hat RPMs on your computer, you can keep working in a virtual console. Press Ctrl+Alt+F2; after a short time, you can run the* `chroot /mnt/sysimage` *command. This brings you to the standard root (/) directory. You can then edit the configuration files of your choice as soon as they are installed.*

Once packages are installed, you've got a lot of work to do. You need to be prepared to do a number of things, including:

- Configuring local or remote printers, as defined in Chapter 20.
- Setting up custom schedules for `cron` and `at` jobs as defined by your exam. We describe how these jobs work in Chapter 14.

- Connecting your system to a NIS or LDAP service. These services are defined in Chapter 23.
- Configuring your system to mount a remote or portable filesystem using the automount daemon, sometimes known as *autofs*. We explain this process in Chapter 7.
- Adding the users and groups as defined by your exam. You may also need to set up quotas. You may also be told to set up the User Private Group scheme, where permissions are configured for collaboration. Both are defined in Chapter 9.
- Naturally, you may need to update RPMs to a later version. In most cases, you'll upgrade; with the kernel RPM, you'll generally want to install so the original kernel is still in place, just in case.
- While installing a Red Hat built kernel RPM automatically updates your booloader. However, there may be other reasons to update bootloaders, such as an extra command your exam might instruct you to add. The bootloader is described in detail in Chapter 11.
- There are a number of runtime parameters associated with the kernel, which you can configure in the `/proc` directory. Some are covered in Chapter 13.

Remember, you have two hours to complete the RHCT Installation and Configuration exam. If it takes a half hour to install Red Hat Enterprise Linux on your computer, you may have an hour and a half left to configure possibly a dozen services or more.

What the RHCT Exam Does Not Cover

The RHCT exam does not include any multiple-choice questions. It does not require you to configure Linux as a network server. It does not require more than the most basic knowledge of network security (as opposed to host computer security).

Preparing for the RHCE Exam

If you're planning to take the RHCE exam, read the previous section. You'll need to pass all RHCT requirements. In the following sections, we describe the additional requirements associated with the RHCE exam.

The prep course for this exam is Red Hat course RH300, RHCE Rapid Track Course. As of this writing, RH300 is a full five-day course. The last day includes five and a half hours for the RHCT exam. Believe me, it is a very intense course. Even many of the most experienced Red Hat users often study course lessons late each night during the course.

As of this writing, the Red Hat website states that candidates without "real-world system administration experience" are not likely to pass the RHCE exam. It is a realistic performance-based lab exam that starts by testing you on all RHCT requirements. It includes requirements to configure a substantial number of network services, diagnose and fix boot problems, reconfigure logical volumes, and more, in a very limited period of time.

The RHCE exam consists of two parts. It begins with a two and-a-half-hour Troubleshooting and System Maintenance exam. It ends with a three-hour Installation and Configuration exam.

The RHCE Troubleshooting and System Maintenance Exam

You'll have two hours to take the RHCE Troubleshooting and System Maintenance exam. In the first hour, you'll have to solve five RHCT-level problems, as defined on that lower-level exam. You need to solve all these RHCT-level problems correctly in order to pass the RHCE exam. You are then faced with a number of RHCE problems in the time that remains. Your overall score on this exam must be at least 80 percent.

If your overall score is less than 80 percent, you may still get the RHCT credential if you solve all the RHCT-level problems within the first hour of this exam. You'll also need the skills described in the following sections.

The Rescue Environment

For the RHCE Troubleshooting and System Maintenance exam, learn how to use the first installation CD as a rescue disk. As described in Chapter 11, that involves knowledge of `linux rescue` mode. Depending on the problem, this mode may allow you unmounted, read-only, or full access to your system, as a root user, in runlevel 1.

Bootloader, Module, and Filesystem Errors

There are a number of issues that can lead to problems during the Linux boot process. The default Red Hat Enterprise Linux bootloader is GRUB, which we describe in Chapter 11.

In the real world there may be a problem with the bootloader. It could be that the kernel file is corrupted; it could be that the Initial RAM disk image is missing. There could even be an error in your bootloader configuration file. Remember, if you have the GRUB password, you can edit various commands in `/boot/grub/grub.conf` before it starts Linux. In this way, you can diagnose a GRUB bootloader problem before even starting the Linux boot process.

During the boot process, Red Hat Enterprise Linux 3 normally relies on modules to load hardware and filesystem drivers. Problems can keep Linux from recognizing your hardware or from mounting directories on appropriate partitions.

We find that Linux filesystem errors are extremely rare, especially when compared to the Microsoft Windows VFAT or NTFS filesystems. However, it does happen. When it happens, it can keep your system from booting, or the boot process may stop with an error. At that point, you need to know the commands that can repair basic Linux filesystems. We describe these tools in Chapter 11.

You can start the boot process in another way, using the rescue mode associated with the Red Hat installation boot disks. This can get you around problems in the Linux boot process, so you can start Linux and fix any files that may not be working.

Once you've started rescue mode, you can look through your Linux system to see what may have gone wrong. For example, your kernel may be missing or corrupted. If Linux can't read your `/etc/fstab` file, it won't know what directories to mount. In fact, if your `mount` command file is corrupted, Linux won't be able to mount your filesystems. If filesystems are corrupted, you'll need to find a way to run `fsck` on the appropriate partition.

If that all works, you could have problems with `/etc/inittab`. If the `id` variable is set to the wrong runlevel, Linux could stop before you have a chance to log in. Errors in the virtual consoles or the `/etc/securetty` file could keep you or your users from logging in.

NOTE *If you have a test computer, where you don't have any valuable data, you can use it to experiment with the boot process. You can create your own boot problems on a test computer. Reboot, and then see how it affects the Linux boot process. Again, do not experiment on a computer that holds important data; this kind of tinkering can easily go astray, erasing all your data.*

TIP *To learn how to troubleshoot the boot process, back up key files such as* `/etc/fstab`*,* `/etc/inittab`*, and* `/boot/grub/grub.conf`*. Make a change to a line in one or more of these files. Reboot your computer, and see what happens. You may need to use* `linux rescue` *mode. Restore the original configuration, and then make a different change.*

Network Diagnosis

Sometimes, you have a network service that isn't working properly. The cause could lie with the configuration of the service, or it could be a problem with one or more of the firewalls described in Chapters 17 and 18. Or you could have one of the network problems described in Chapter 16.

Checking Your Network

Most network problems are physical. Fortunately, the RHCE exam assumes that the physical components of the network are in good working order. Thus, checking your network becomes an exercise in tracing data. As we discussed in Chapter 16, if networking is properly configured, you should be able to `ping` your local computer, and you should see your network card in the `ifconfig` output. However, not all network tests will work; for example, your attempts to `ping` a server could be blocked by a firewall.

Checking Firewalls

By default, when you install Red Hat Enterprise Linux, you get an `iptables` firewall on your computer. If you're installing Linux on a computer that's on a LAN that is already protected by a firewall, you may not need that extra layer of protection. And even the default medium-security firewall can block shared NFS directories.

Your computer could be configured with TCP Wrappers firewalls that can block many network services. Network traffic can be let in through `/etc/hosts.allow`; traffic can be blocked through `/etc/hosts.deny`.

Finally, individual services can be configured with their own firewalls. For example, you can block access to a WU-FTP server to specific users in `/etc/ftpaccess`.

Checking Individual Services

You may also need to diagnose problems with individual network services, including those that are cited in the Red Hat Exam Prep Guide. As of this writing, they include Apache (regular and secure hosts), Samba, NFS, FTP, Proxy, SMTP, incoming e-mail (IMAP, IMAPS, and POP3), SSH, and DNS.

We've shown you how to diagnose problems with each of these services in the applicable chapters in this book.

Reconfigure Logical Volumes

While it's slightly faster to configure Logical Volumes during the installation process, you may not have that opportunity during the Troubleshooting and System Maintenance exam.

We show you how to add, remove, and resize logical volumes in Chapter 7.

The RHCE Installation and Configuration Exam

This may be where you feel the most pressure during the RHCE exam. If you don't have a lot of practice configuring services, you may struggle to complete all the tasks on this exam in time.

On this exam, installing Linux is the easy part. You also need to configure several services. If you forget to install the software associated with a service during installation, you'll need to install it later. For many candidates, the three hours allocated for this exam is not enough time.

Tip *Red Hat has recently added one more criterion for passing the RHCE exams. The RHCE Installation and Configuration exam includes RHCT- and RHCE-level components. To become an RHCE, you now also need a score of at least 70 percent on both the RHCT and RHCE portions of the RHCE Installation and Configuration exam.*

What You Can Expect

On the RHCE Installation and Configuration exam, you'll be asked to install Red Hat Enterprise Linux 3 on your computer. You'll need to meet the requirements described earlier for the RHCT Installation and Configuration exam.

You may also need to set up several websites, start a secure FTP server limited to certain users, create a firewall, share NFS directories with certain computers, limit access to shared Samba directories, or set up a DNS and a DHCP server with certain names and IP addresses.

You could also have to configure a web proxy server such as Squid, set up an outgoing SMTP e-mail server such as sendmail, or install incoming e-mail servers such as IMAP, IMAPS, and POP3. Remote access can be secured with a service such as SSH.

Time is of the essence. If you know how to configure services in text mode, you may save enough time to configure one more service properly. On the other hand, if exam pressure makes you forget how to configure a service at the command line, some of the Red Hat graphical tools may be a lifesaver.

In many cases, there may be more than one way to complete a task. There is no "right" way to do something; for example, if you want to block access from a specific network to a Telnet server, you could do so with the proper `iptables` command or the right commands in `/etc/hosts.deny`.

Configuration during Installation

On this part of the RHCE exam, you get to install Linux on a computer—no big secret there. However, what you do during the installation process can save or cost you the precious minutes that you may need to complete the tasks on this exam.

Unless you need to configure LVM volumes, it is faster to install Red Hat Enterprise Linux using text mode. The computer that you're using may not be the fastest one available; installing Linux in graphical mode does take extra time. If the installation files are available over a faster network, such as Fast Ethernet, install Red Hat Enterprise Linux over that connection. You may get Linux installed in less than half the time of an installation from CDs.

Read through the exam. Make careful notes on the desired partitions, if any. Although you can configure additional partitions after Linux is installed, the process is at best time-consuming and at worst fraught with risks.

Make a note of the services you'll need. When you select packages, be sure to include those required to support your services. If necessary, include appropriate documentation packages.

If you're asked to protect your system from access, you may be able to configure a firewall during the installation process. If you learn how to let other services through your firewall during Red Hat Enterprise Linux installation, that's one less service you'll have to configure later.

Note *Once Anaconda starts installing the hundreds of Red Hat RPMs on your computer, you can keep working in a virtual console. Press Ctrl+Alt+F2; after a short time, you can run the* `chroot /mnt/sysimage` *command. This brings you to the standard root (*`/`*) directory. You can then edit the configuration files of your choice as soon as they are installed.*

Configuring Network Services Quickly

Generally, you have two choices when you configure network services. You can edit the configuration files directly, or you can edit them using a GUI tool. Your choice depends on your level of knowledge of the associated configuration file.

In other words, if you know a configuration file well, you can save time by editing that file directly. If you're less comfortable with that service, or nerves make you forget how to configure services during this exam, you may save time by using the appropriate `redhat-config-*` tool.

Tip *Don't rely on GUI tools during the Red Hat exams. Although one or two may not slow you down too much, we believe that complete reliance on GUI tools would make it extremely difficult to finish the exam on time.*

Whatever you choose, remember to activate the service, now and for the next time you boot that Linux computer. To activate the service now, it's usually enough to run the `service` *`daemonname`* `start` command. However, to ensure that the service starts the next time Linux reboots, you need to run a command such as

```
chkconfig        --level 235 daemonname on.
```

Otherwise, you may not get full credit for the work that you've done to configure that service.

Don't overdo what you configure. For example, if you're told to create two virtual websites, each with a single web page, keep it simple. Unless otherwise directed, you can save a simple text message as `index.html` in the appropriate `DocumentRoot`, such as :

```
This Website works!
```

Configuring Network Service Security

You may need to configure different levels of security on different services during the RHCE exam. Red Hat Enterprise Linux includes user-based and host-based security. If you've read through this book, you'll know that you can configure security at four basic levels:

- With firewall commands such as `iptables` or related tools such as `redhat-config-securitylevel`.

- Using TCP Wrappers to limit access to daemons via the `/etc/hosts.allow` and `/etc/hosts.deny` files.
- Using PAM modules to regulate access; with the right configuration files, this can help you limit access to specific users.
- With appropriate commands within specific service-configuration files. Many support both host- and user-based security commands.

We've described what you can do to secure individual services in a number of different chapters.

Configuring Kickstart

There is one more skill that does not fit in any of the other categories. Linux administrators may need to install Red Hat Enterprise Linux on a number of workstations simultaneously. Red Hat's tool for automated installation is Kickstart. As we describe in Chapter 5, you can set up a Kickstart file to install Red Hat Enterprise Linux 3, ideally from a network source. You can set up a customized Kickstart configuration file on a local floppy or on a network server, possibly with the help of a DHCP server.

Summary

Red Hat Certified Engineers (RHCE) and Red Hat Certified Technicians (RHCT) are respected in the Linux community. They have passed a hands-on exam that has measured their skills in real-world situations. The material on the Red Hat exams is challenging; Red Hat has come up with a series of prerequisites that you should know even before studying for either exam.

While you don't need to know all the prerequisites to have a chance at passing the RHCE exam, it is a good measure of your basic skills. The prerequisites include basic hardware knowledge, a basic understanding of the `vi` editor, and a strong grasp of the Filesystem Hierarchy Standard. You should also know a number of basic bash shell commands as well as LPD and CUPS print commands, and you should understand what is required to configure the shell. Other prerequisites address basic password and file security, in addition to system administration skills. Finally, you need a good grounding in TCP/IP networking, IP addressing, as well as network services and security.

The RHCT exam is a direct subset of the RHCE exam. In other words, if you take the RHCE exam, you'll have to solve a number of RHCT problems. Both RHCE and RHCT exams include two parts: the Troubleshooting and System Maintenance exam and the Installation and Configuration exam. For Troubleshooting and System Maintenance, you have to solve all RHCT-level problems on either exam.

On the Installation and Configuration part of both exams, you'll need to install and configure Linux. Time is of the essence on this exam, especially with the list of services, users, and files that you may need to configure. On the RHCT exam, you'll need to configure Linux as a workstation, with connections to an existing network. On the RHCE exam, you'll also have to configure a number of Linux network services and more.

Now we're coming into the home stretch for this book.Next, we'll look at Part VIII, where we learn how to manage the X Window in Red Hat Enterprise Linux. This starts in Chapter 29 with a detailed review of how to configure basic X Servers and X Clients. We'll examine configuration tools and the files they affect in detail. Then we'll look at how this can work for remote graphical applications.

Part 8

Window Management

In this part, you will learn how to:

Chapter 29

Managing X Servers and X Clients

NEWER LINUX USERS OFTEN prefer a graphical user interface (GUI). If they're not administrators, they don't need the flexibility of the command line. They do need optimized graphics to design airplanes, create movies, chart statistical data, and perform other tasks. Some are regular consumers who want an easy transition from another operating system. The two most common GUIs are GNOME and KDE (see Chapter 30).

While most veteran Linux administrators prefer the command-line interface, they should recognize that many users have a legitimate need for the GUI. To this end, Red Hat Enterprise Linux includes the X Client and X Server system developed by the XFree86 project (`www.xfree86.org`). Linux GUIs use this client-server structure.

You may have already configured the X Window and installed GNOME and/or KDE when you installed Red Hat Enterprise Linux. As long as you've installed the basic X packages, you can use the basic Red Hat Display Settings (`redhat-config-xfree86`) tool to configure the X Window on your computer.

The critical X Window configuration file is `XF86Config`, in the `/etc/X11` directory. It includes a number of sections that we'll analyze in detail. There are several other significant X Window configuration files that can help you customize your system. This chapter covers the following topics:

- Using the basic configuration tools
- Understanding the configuration files
- Configuring Remote X Access
- Troubleshooting the X Window

Using the Basic Configuration Tools

When you configure the X Window on your computer, you must configure several parts of your computer. Not only do you need to configure graphics, but also any input device that might interact with a graphical screen. These components include the following:

- Monitors with specifications for horizontal and vertical frequency, resolution, and refresh rates
- Video cards with a specified amount of memory
- A mouse or other pointing device for a GUI
- Keyboards to support a GUI

This data is documented in `/etc/X11/XF86Config`. You could edit this file directly. In fact, we'll review this file in detail later in this chapter. Unfortunately, the language within the file is a little obscure. Thus, most people use an X Window configuration tool to help with the process.

The X Window configuration tool is `redhat-config-xfree86`. Red Hat no longer includes three other formerly popular configuration tools, `xf86config`, `Xconfigurator` and `XF86Setup`.

If Linux can detect your hardware, there is one simple alternative for creating an X Window configuration file: the `X -configure` command.

As of this writing, the people behind Fedora have replaced the XFree86 project software with servers from the X Project (`www.x.org`). Red Hat has stated that the Fedora project will be the test bed for future Red Hat software. Therefore, we believe that the information in this chapter will change significantly for Red Hat Enterprise Linux 4.

X Window RPMs

Normally, if you want to install more packages, you just start the `redhat-config-packages` utility described in Chapter 10. This opens the Package Group Selection screen (see Chapter 3), where you can select different package groups. But that utility doesn't work unless you've already installed a GUI.

If you need to install X Window RPMs, use the `rpm` command (refer to Chapter 10) to install the packages in the X Window System package group, known in the `comps.xml` file as the `base-x` group. You can find the `comps.xml` file on the first Red Hat Enterprise Linux installation CD, in the `/RedHat/base` directory.

X Window RPMs may not be enough. You'll need more for a GUI desktop. As explained in Chapters 3 and 4, the GNOME and KDE Desktop Environment package groups require a different set of RPM packages.

Red Hat Display Settings (*redhat-config-xfree86*)

The tool for configuring the X Window on Red Hat Enterprise Linux is the Red Hat Display Settings tool, which you can open with the `redhat-config-xfree86` command. In most cases, you can even run it from the standard command-line interface; it probes your monitor and graphics card and opens the basic dialog boxes with a VESA interface.

NOTE *VESA is the basic graphical interface developed by the Video Electronics Standards Association. The associated generic settings are also known as Super VGA.*

The `redhat-config-xfree86` command detects your hardware. The associated Display Settings tool includes sections for the overall display, the monitor, and the video card.

Detecting Hardware

Before `redhat-config-xfree86` opens the Display Settings tool, it runs the `ddcprobe` command. You can run this command yourself. Figure 29.1 illustrates the effect on my desktop computer.

Figure 29.1
ddcprobe detects a monitor and video card.

```
[root@Enterprise3d root]# ddcprobe

Videocard DDC probe results
Description:  Intel Corporation i810 Graphics Controller
Memory (MB):  1

Monitor DDC probe results
ID: SAM413b
Name: Samsung SyncMaster 955df
Horizontal Sync (kHZ): 30-85
Vertical Sync (HZ)  : 50-160
Width (mm): 360
Height(mm): 270
[root@Enterprise3d root]#
```

The Overall Display

When the Display Settings tool opens, you'll see a window similar to Figure 29.2. The Display tab allows you to select a resolution and color depth. The available settings are based on what the video card can do and reflect the limits of the monitor.

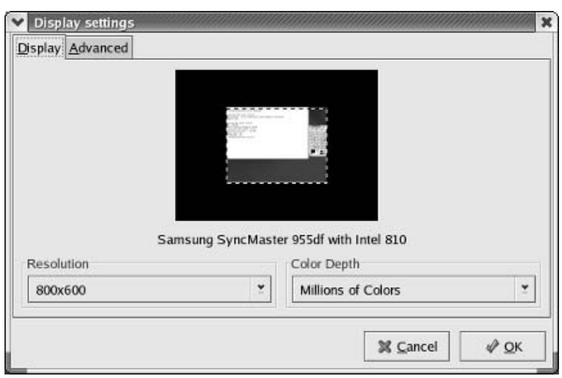


Figure 29.2
The Display tab

The top of the screen illustrates open GUI applications on your computer. This can help you get a feel for how the applications will look on your monitor. If you change the Resolution setting, the dotted lines around the applications change as well.

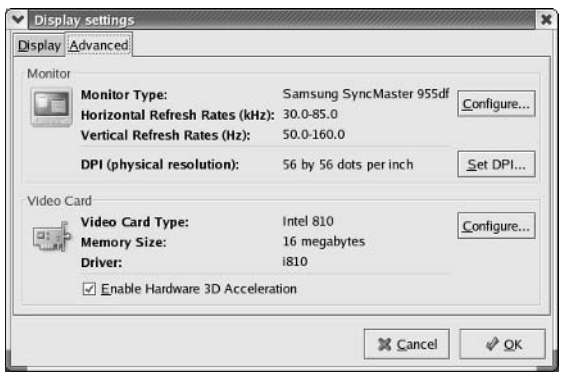
Resolution represents the number of dots that your video card sends to your monitor. The number is in horizontal × vertical format; 800 × 600 resolution means that there are 800 dots across in the horizontal plane and 600 dots in the vertical plane. For a list of other available resolutions, click the Resolution drop-down arrow.

The Color Depth setting represents the number of colors available for each dot. For example, 16-bit color means you can have any of 216 = 65,536 colors in each dot. For a list of other available color depths, click the Color Depth drop-down arrow.

The Video Card

Back in the Display Settings window, click the Advanced tab. As you can see in Figure 29.3, the lower half of this screen includes your Video Card settings.

Figure 29.3
The Advanced tab



You can further configure the video card. Click Configure in the Video Card section of the Advanced tab. You're taken to the Video Card Settings window, shown in Figure 29.4.

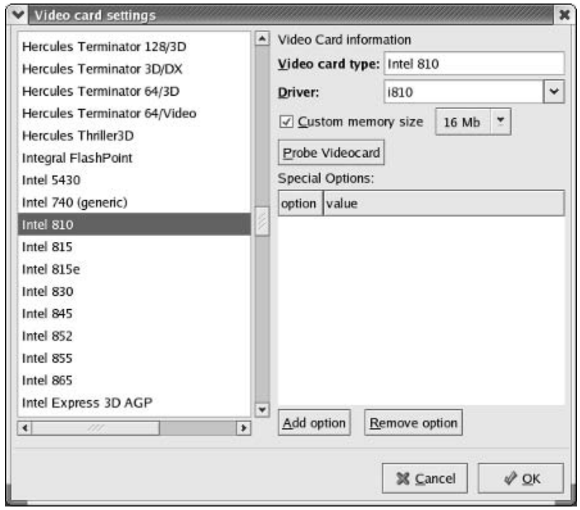
You can select from well over 600 makes and models of video cards. When you do, the Video Card Type and Driver appear automatically in the upper-right corner. In some cases, you'll see special option commands in the lower-right corner.

Alternatively, you can try clicking the Probe Videocard button. In many cases, the Display Settings tool can detect your video card and select the appropriate drivers automatically.

If you don't see a video card that matches your make and model, you have several options:

- Select the VESA Driver (Generic) card type. This assigns standard settings (SVGA) with the `vesa` driver that should work for most video cards built in the past several years.
- Select the Unsupported VGA Compatible card type. This assigns the `vga` driver to your system.
- Select Custom (at the top of the list). You may add a Linux driver from the video card manufacturer or a third party to the video modules directory, `/usr/X11R6/lib/modules/drivers`.

FIGURE 29.4
Video card settings



Whether you use a model-specific or a generic driver, be sure to check the Custom Memory Size setting. Revise it if it does not match the actual amount of graphics memory on your video card.

Several video cards allow you to configure various options, such as acceleration, depth, and orientation. You can use the Add Option button for this purpose. Make your selections, and click OK to continue.

NOTE *If you want more information on the options available, get the make and model of your video card. Navigate to* `www.xfree86.org/4.3.0/RELNOTES.html`*, and look for the Video Drivers section. You'll see links for the make and model of your video card. Video card–specific* `XF86Config` *file options are also documented here.*

When you return to the Advanced tab of the Display Settings window, look at the Enable Hardware 3D Acceleration check box. If your video card has this capability, you should be able to activate the check box. Now let's look at your monitor.

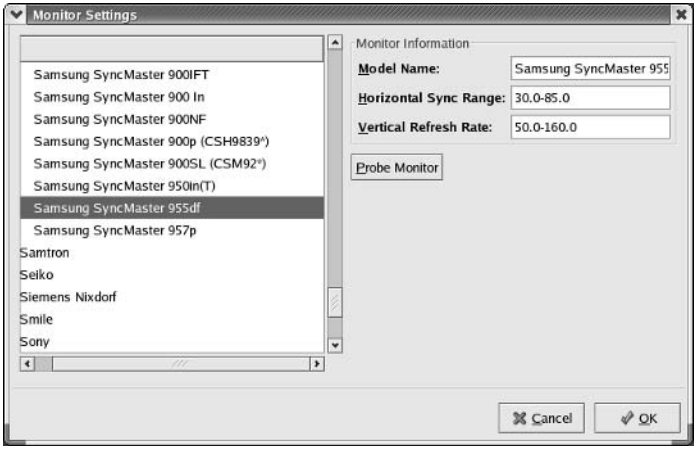
THE MONITOR

The Display Settings tool also allows you to configure X Server settings for your monitor. Once again, open the Advanced tab of the Display Settings window, and then click the Configure button to open the Monitor Settings window, shown in Figure 29.5.

You can use the Display Settings tool to configure monitors from well over 100 manufacturers. If you see the manufacturer of your monitor, click the arrow adjacent to the name. This should open a selection of models made by that manufacturer.

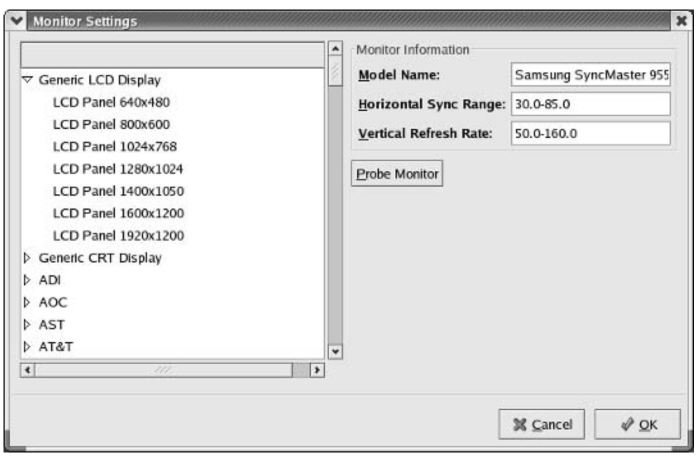
Alternatively, you may find an exact match when you click the Probe Monitor button.

FIGURE 29.5
The Monitor Settings window



If you can't find an exact match, a large selection of generic monitors is available; part of the current list is shown in Figure 29.6. As you can see, the Generic settings are divided into LCD and CRT groups. LCD monitors were originally associated with iquid Crystal Displays; it now is associated with flat panel and laptop monitor screens. CRT monitors are based on the more traditional Cathode Ray Tube, which relies on an electron beam some distance behind the screen.

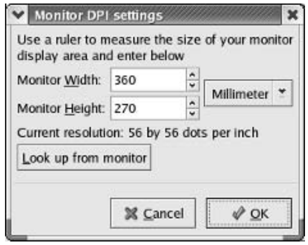
FIGURE 29.6
Generic monitor settings



While you can customize the Horizontal Sync Range and Vertical Refresh Rate of your monitor, be careful. Check the documentation for your monitor. If the numbers you select are too large, you may exceed the capabilities and destroy your monitor. Although many monitors include protection against such overloads, why risk blowing out your new flat-panel or laptop screen?

When you complete your settings, click OK to return to the Advanced tab. Click the Set DPI button to open the Monitor DPI Settings window, shown in Figure 29.7.

FIGURE 29.7
The Monitor DPI Settings window



As you can see, you can customize the size of the picture on your monitor screen. Using the drop-down box, you can set the width and height of your monitor in millimeters or inches. If you click the Look Up From Monitor button, you get the current settings. Make any desired changes, and click OK.

When you click OK in this window, the Display Settings tool saves your changes to `/etc/X11/XF86Config`. Your settings take effect the next time you log into a GUI on this computer.

Auto X Configure

If the Display Settings tool is not to your liking, there is one more option. If the XFree86 Server can detect your video card and monitor, there's a simple alternative. Try the following command:

```
# X -configure
```

If successful, it'll create the `XF86Config.new` file in the local directory. Back up your current `/etc/X11/XF86Config` file. You may be able to make additional changes to your `XF86Config` file, as described later in this chapter. When you're ready, overwrite your `/etc/X11/XF86Config` file with `XF86Config.new`. Run the `startx` command to test the result.

NOTE *In my experience, the* `X -configure` *command, when run in Linux inside a VMWare virtual machine, causes the system to "black out."*

switchdesk

In Red Hat Enterprise Linux, GNOME is the default desktop. If you use a variety of desktops, the `switchdesk` utility provides an easy way to start a different GUI. If you run `switchdesk` from inside a GUI, you'll see something similar to the Desktop Switcher window shown in Figure 29.8.

FIGURE 29.8
The Desktop Switcher window

The Desktop Switcher window shows your installed GUIs; you can use it to switch between installed desktop environments such as GNOME and KDE.

You can also use `switchdesk` from the command-line interface. It's simple; for example, if you want to make KDE your default desktop, run this command:

```
# switchdesk KDE
```

It takes effect the next time you log into the Linux GUI.

Linux GUI Desktops

Several Linux GUI desktops are available. Some of the major options can be used by `switchdesk`:

- GNOME is the default Red Hat Enterprise Linux GUI desktop; the acronym stands for the GNU Network Object Model Environment.
- KDE is the other major GUI desktop; the acronym stands for the K Desktop Environment.
- The fvwm (and fvwm95) window manager was the standard Red Hat GUI before GNOME and KDE. Because it requires only a small amount of memory, it suited the time when RAM was more expensive.
- Enlightenment is perhaps the most configurable of the major Linux GUI window managers.
- The twm window manager is very basic; on Red Hat Enterprise Linux, it includes one console screen. It also serves as a failsafe desktop environment, with minimal tools and programs.
- The WindowMaker window manager is designed to be more intuitive; it looks vaguely like the GUI for the NeXTStep operating system.

Changing the Display Manager

A display manager is the login manager, which provides a graphical look and feel to users when they log into your Linux computer. Three major options are available for display managers. Two are associated with GNOME and KDE; the third is a generic X Window display manager.

You can select your preferred display manager in `/etc/X11/prefdm`. The key variable is about 25 lines into this file:

```
preferred=
```

Depending on your preferred display manager, you can set the preferred variable to *one* of the following lines:

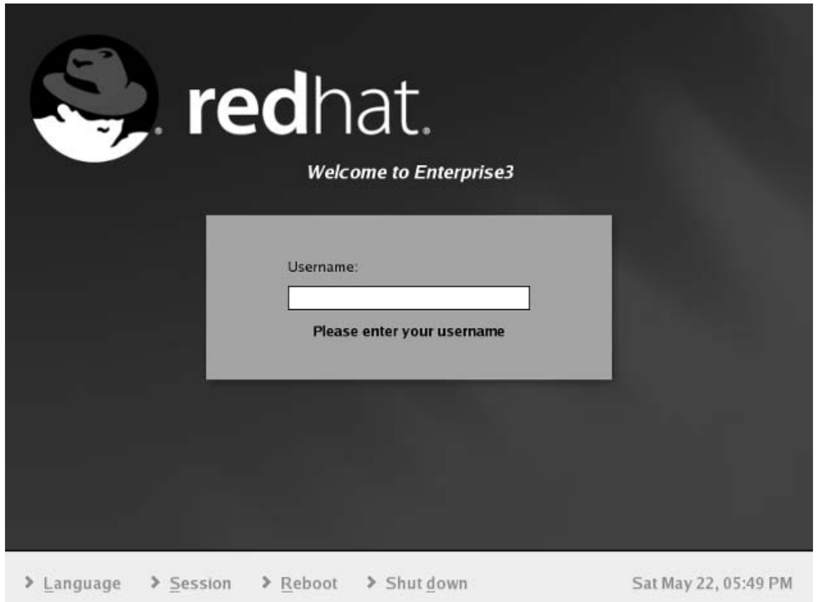
```
preferred=gdm
preferred=kdm
preferred=xdm
```

These refer to the GNOME Display Manager, the KDE Display Manager, and the X Display Manager, respectively. Let's examine each in turn.

The GNOME Display Manager

The GNOME Display Manager is shown in Figure 29.9.

Figure 29.9
The GNOME Display Manager



Besides the straightforward login interface (which prompts you for a password), there are four menus:

Language If you've installed the appropriate language packages, you can click Language and select that language for your session.

Session This opens a menu that allows you to select from available desktops.

Reboot This prompts for confirmation before rebooting the computer.

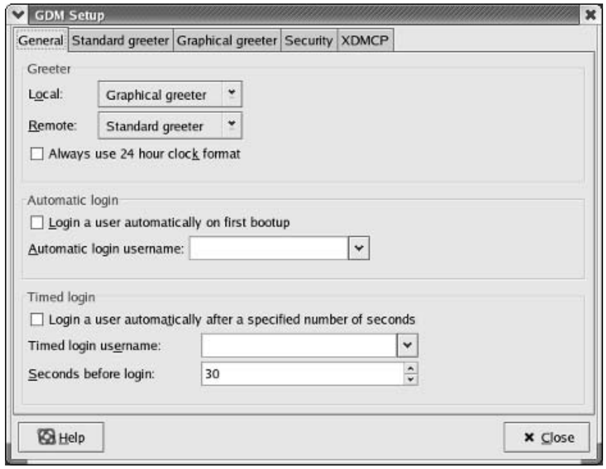
Shutdown This prompts for confirmation before shutting down the computer.

You can configure the GNOME desktop manager. Open up a GUI, and run the `gdmsetup` command to open a menu with five tabs, as shown in Figure 29.10.

These tabs can help you customize the GNOME Display Manager in several ways:

- The General tab allows you to configure basic local and remote login parameters.
- The Standard Greeter tab gives you control over the look and feel of this interface, normally used for remote graphical connections.

Figure 29.10
The GDM Setup window



- The Graphical Greeter tab gives you a choice of several themes for the graphical `gdm` interface. You may be able to install new themes as they are developed by Red Hat, the GNOME project, or a third party such as Ximian.
- The Security tab lets you regulate root and remote logins, as well as available login menus.
- The XDMCP tab allows you to configure how this display manager communicates with remote users. XDMCP is the X Display Manager Control Protocol.

If you want to customize a KDE login interface, you can configure the KDE Display Manager through the KDE Control Center Login Manager setting. No equivalent configuration tool is available for the X Display Manager (`xdm`); however, it includes configuration files in the `/etc/X11/xdm` directory.

The KDE Display Manager

You can also configure the KDE Display Manager, as shown in Figure 29.11. This manager also includes a straightforward login interface and several options.

Session Type Allows you to select from available desktops.

Go Sends your typed-in username and password for verification.

Clear Erases entries in the Username and Password text boxes.

Menu Allows you to restart the X Server.

Shutdown Opens a window that allows you to send the Turn Off Computer (`poweroff`) or Restart Computer (`reboot`) commands.

FIGURE 29.11
The KDE Display Manager

THE X DISPLAY MANAGER

Finally, you can configure the X Display Manager, as shown in Figure 29.12. This is the most straightforward login interfaces; all you can do from this screen is log into this computer.

FIGURE 29.12
The X Display Manager

DEFINITIONS

You should keep in mind a number of definitions when talking about the X Window and the GUI. Several of these terms are closely related and are used interchangeably, which can be confusing.

Display manager A graphical interface for logins. Common display managers include the X Display Manager (`xdm`), the GNOME Display Manager (`gdm`), and the KDE Display Manager (`kdm`).

Desktop A window manager integrated with a series of tools and programs. The two most common desktops are GNOME and KDE. The GNOME desktop does not have to include the GNOME window manager. For example, older versions of Red Hat Enterprise Linux configured an Enlightenment window manager on a GNOME desktop.

Graphical user interface (GUI) A graphical interface through which a user can interact with a computer. A combination of an X Server and X Clients.

Window manager A specialized X Client that controls the look and feel of and the interface to windows in a GUI.

X Client An application that is run within a GUI; it can be run from the local or from a remote computer.

X Server The drivers and programs that create the GUI on the local computer.

Understanding the Configuration Files

Several important configuration files and executable programs are related to the Linux X Window system. Most of you know the command that starts the X Windows from a command-line interface:

```
# startx
```

This program refers to other configuration files and programs, in the `/etc/X11/xinit` and `/usr/X11R6/bin` directories. The `/etc` files can be customized for individual users, as hidden files in their home directories.

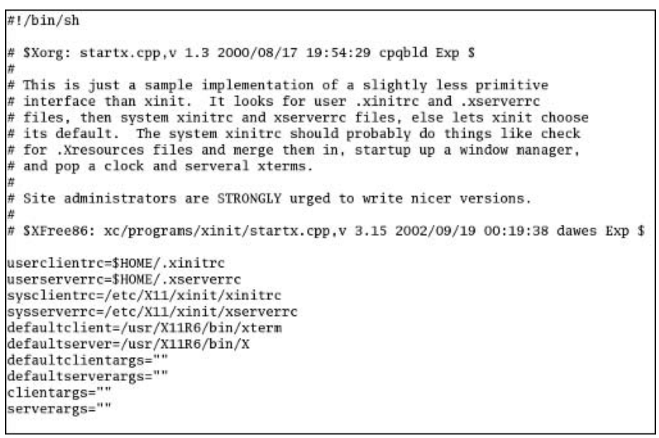
By far, the most important X Window configuration file is `XF86Config` in the `/etc/X11` directory; we'll discuss that file in some detail later in this section.

startx

There are three basic ways to get into the Linux GUI. You can edit the `id` variable in `/etc/inittab` to start in runlevel 5 when you boot Linux, or you can go into runlevel 5 from the text console with the `init 5` command. (More information on `/etc/inittab` and `init` is available in Chapter 11.) Either of these methods brings you to one of the graphical login interfaces described earlier.

A third method is to run the `startx` command. This is actually an executable file in the `/usr/X11R6/bin` directory. You can open the `startx` script in any text editor. The start of this file is shown in Figure 29.13.

FIGURE 29.13
The startx file



```
#!/bin/sh

# $Xorg: startx.cpp,v 1.3 2000/08/17 19:54:29 cpqbld Exp $
#
# This is just a sample implementation of a slightly less primitive
# interface than xinit.  It looks for user .xinitrc and .xserverrc
# files, then system xinitrc and xserverrc files, else lets xinit choose
# its default.  The system xinitrc should probably do things like check
# for .Xresources files and merge them in, startup up a window manager,
# and pop a clock and serveral xterms.
#
# Site administrators are STRONGLY urged to write nicer versions.
#
# $XFree86: xc/programs/xinit/startx.cpp,v 3.15 2002/09/19 00:19:38 dawes Exp $

userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/etc/X11/xinit/xinitrc
sysserverrc=/etc/X11/xinit/xserverrc
defaultclient=/usr/X11R6/bin/xterm
defaultserver=/usr/X11R6/bin/X
defaultclientargs=""
defaultserverargs=""
clientargs=""
serverargs=""
```

As you can see, this script includes several variables. It first looks for `.xinitrc` and `.xserverrc` files in the home directory of the requesting user. If these files aren't available, it uses defaults in the `/etc/X11/xinit` directory.

NOTE *The* `/etc/X11/xinit/xserverrc` *file does not exist by default on Red Hat Enterprise Linux 3 systems; instead, the* `startx` *command starts the X Server in the first available graphical console, with the* `X :0` *command.*

The `defaultclient` and `defaultserver` are the default X Client and the default X Server; the default `xterm` client is used if you use `switchdesk` to make twm your default desktop. The other variables are intentionally left empty; if you're comfortable with programming code, you'll be able to see how these variables are assigned.

/etc/X11

The `/etc/X11` directory contains a number of important configuration files and directories. Table 29.1 describes each of the files and subdirectories.

TABLE 29.1: */ETC/X11* **FILES AND DIRECTORIES**

File or Directory	Description
`applink`	A directory with links to applications that appear in a GUI Start menu.
`desktop-menus`	A directory with settings for various default GUI menus.
`fs`	A directory with the Font Server configuration.
`gdm`	A directory with GNOME Display Manager configuration files.
`lbxproxy`	A directory for remote clients that want to use the low-bandwidth extension to the X Server (LBX).
`prefdm`	A file that selects the preferred display manager.
`proxymngr`	A directory with configuration for use with proxy managers.
`serverconfig`	A directory for X Server configuration settings.
`starthere`	A directory with basic X desktop settings.
`sysconfig`	A directory with a `gnome-lokkit` configuration file.
`twm`	A directory with the twm configuration file, `system.twmrc.`
`X`	A file linked to the X Server application.
`xdm`	A directory with X Display Manager configuration files.
`XF86Config`	The main X Server configuration file.
`xinit`	A directory with default X configuration files called by `startx`; used if equivalent files are not available in the applicable home directory.
`xkb`	A directory for keyboard configuration.
`Xmodmap`	A default configuration file for keyboards.
`Xresources`	A configuration file that calls fonts for the login screen.
`xserver`	A directory with a `SecurityPolicy` configuration file.
`xsm`	A directory that configures the X session manager.

Local Configuration Files

You can set up X Window configuration files in users' home directories. As you've seen earlier, `startx` looks for two of them for settings to start the Linux X Window: `~/.xinitrc` and `~/.xserverrc`. The dot hides these filenames in your home directory.

NOTE *As described in Chapter 6, you can view the hidden files in any directory with the* `ls -a` *command.*

As described earlier, Red Hat does not use the `xserverrc` file. Thus, the key configuration file (if used) is `~/.xinitrc`, which also calls several other files in the home directory.

NOTE *Remember, the tilde (~) represents the current user's home directory.*

The other key files are `~/.Xclients` and `~/.Xclients-default`, which `switchdesk` modifies so `startx` knows the desktop you want. If you're interested in how these files work, read them for yourself. Use the `switchdesk` command as described earlier to set a different default desktop and see what that does to `~/.Xclients-default`. Finally, the `~/.Xresources` file sets default color and dimensional parameters for the `emacs`, `xterm`, and Seyon clients.

XINITRC

When the `startx` command starts your X Server, it needs to call up fonts, keyboard settings, and default X Clients.

The `xinitrc` file is an executable shell script. You can use the default in the `/etc/X11/xinit` directory, or you can customize it, change its name to `.xinitrc`, and store it in your own home directory. The following is a detailed analysis of the default `xinitrc` file:

```
#!/bin/sh
# (c) 1999-2002 Red Hat, Inc.
userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
userxkbmap=$HOME/.Xkbmap

sysresources=/etc/X11/Xresources
sysmodmap=/etc/X11/Xmodmap
sysxkbmap=/etc/X11/Xkbmap
```

These first lines represent the other configuration files needed through the rest of the script. You'll see in a moment that if the `user*` variable files aren't available, `xinitrc` uses just the `sys*` files.

```
# merge in defaults
if [ -f "$sysresources" ]; then
    xrdb -merge "$sysresources"
fi

if [ -f "$userresources" ]; then
    xrdb -merge "$userresources"
fi
```

These lines start by applying the `$sysresources` file, `/etc/X11/Xresources`. If there's a valid `$userresources` file (`~/.Xresources`), the settings from each file are combined.

```
# merge in keymaps
if [ -f "$sysxkbmap" ]; then
    setxkbmap `cat "$sysxkbmap"`
    XKB_IN_USE=yes
fi

if [ -f "$userxkbmap" ]; then
    setxkbmap `cat "$userxkbmap"`
    XKB_IN_USE=yes
fi
```

These lines serve the same purpose as the previous stanzas, except they apply to the noted Keyboard Map files, based in `Xkbmap`. However, that file doesn't normally exist in Red Hat Enterprise Linux and is therefore ignored. In `xinitrc`, this is followed by a stanza related to a Sun Microsystems X Server, which Red Hat does not use and therefore also ignores.

```
if [ -z "$XKB_IN_USE" -a ! -L /etc/X11/X ]; then
     if grep '^exec.*/Xsun' /etc/X11/X > /dev/null 2>&1 && [ -f
  ➥/etc/X11/XF86Config ]; then
          xkbsymbols=`sed -n -e 's/^[    ]*XkbSymbols[
      ➥   ]*"\(.*\)".*$/\1/p' /etc/X11/XF86Config`
          if [ -n "$xkbsymbols" ]; then
              setxkbmap -symbols "$xkbsymbols"
              XKB_IN_USE=yes
          fi
     fi
fi
```

As you're not using the Solaris operating system, you don't need to be concerned about the previous stanza. This is followed by:

```
# xkb and xmodmap don't play nice together
if [ -z "$XKB_IN_USE" ]; then
    if [ -f "$sysmodmap" ]; then
        xmodmap "$sysmodmap"
    fi

    if [ -f "$usermodmap" ]; then
        xmodmap "$usermodmap"
    fi
fi

unset XKB_IN_USE
```

This stanza checks for an Xmodmap file in /etc/X11/xinit or a hidden version in your home directory. If it exists, it's used in place of the aforementioned Xkbdmap file. But the Xkbdmap file doesn't normally exist in Red Hat Enterprise Linux.

```
# run all system xinitrc shell scripts.
for i in /etc/X11/xinit/xinitrc.d/* ; do
    if [ -x "$i" ]; then
        . "$i"
    fi
done
```

This stanza runs basic shell scripts in the noted directory, /etc/X11/xinit/xinitrc.d. These scripts can include files such as xinput and xmbind, which are described later.

```
if [ -f $HOME/.Xclients ]; then
    [ -x /usr/bin/ssh-agent -a -z "$SSH_AGENT_PID" ] && \
        exec ssh-agent $HOME/.Xclients || \
        exec $HOME/.Xclients
elif [ -f /etc/X11/xinit/Xclients ]; then
    [ -x /usr/bin/ssh-agent -a -z "$SSH_AGENT_PID" ] && \
        exec ssh-agent /etc/X11/xinit/Xclients || \
        exec /etc/X11/xinit/Xclients
else
```

These commands check for default clients in the Xclients file. They also set up an authentication agent for SSH, if previously configured. See Chapter 18 for more information.

```
        xclock -geometry 100x100-5+5 &
        xterm -geometry 80x50-50+150 &
        if [ -x /usr/bin/netscape -a -f /usr/share/doc/HTML/index.html ]; then
                netscape /usr/share/doc/HTML/index.html &
        fi
        if [ -x /usr/X11R6/bin/fvwm2 ]; then
                exec fvwm2
        else
                exec twm
        fi
fi
```

These commands set up default clients if no Xclients file is available. You may note that this stanza includes Netscape, which is no longer included with the Red Hat Enterprise Linux CDs. This stanza also uses xclock, a generic Linux GUI clock, and xterm, a generic command-line interface window.

You can also create your own .xinitrc file in your home directory. Make sure to use the appropriate chmod command to make that file executable. For example, you could add the following information to .xinitrc:

```
#!/bin/bash
xclock &
```

```
xterm &
gimp &
exec twm
```

This file starts with `#!/bin/bash`, which assumes that the commands that follow are based on the bash shell. The remaining commands start the standard Linux graphical clock (`xclock`), a basic terminal command-line interface (`xterm`), and The GIMP (`gimp`), which is the GNU Image Manipulation Program. Finally, the code starts a simple window manager interface known as twm (`twm`). The result starts the programs with twm, as shown in Figure 29.14. This even overrides any settings you have in the `~/.Xclients-default` file.

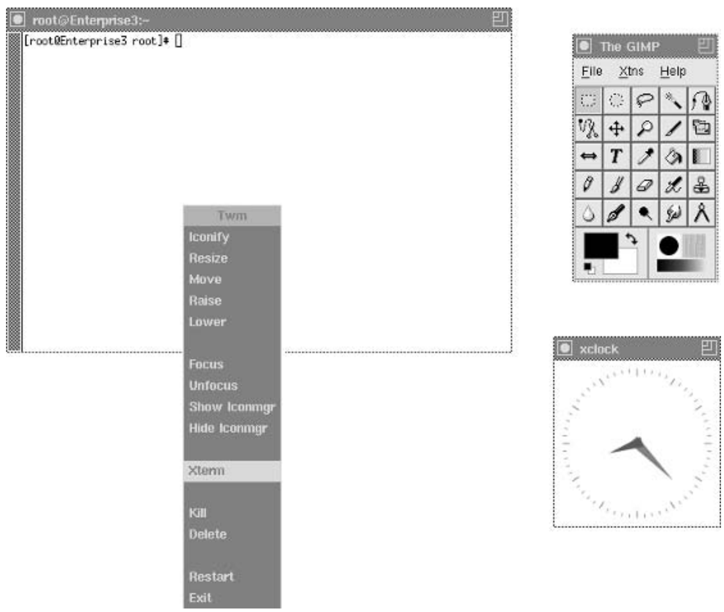


FIGURE 29.14 What you can do with.xinitrc

XRESOURCES

There is normally an `.Xresources` file in users' home directories, as well as a default `Xresources` in `/etc/X11`. The default file goes through a series of steps to find your preferred desktop. Generally, if GNOME or KDE is not available, `Xresources` looks for a `.wm_style` file in your home directory that may call for an older window manager.

But these are details; the standard `.Xresources` file in your home directory sets a color scheme for basic X Clients in your GUI.

XF86Config

The `/etc/X11/XF86Config` file contains the main configuration settings for the X Server. Whenever you start a Linux GUI, the basic settings for resolution, pitch, graphics drivers, monitors, keyboards,

and mice or other pointing devices are configured through this file. This file includes several major sections described here.

The first line in a file tells you if XF86Config was created through Anaconda:

```
# XFree86 4 configuration created by pyxf86config
```

or through redhat-config-xfree86

```
# XFree86 4 configuration created by redhat-config-xfree86
```

NOTE *Prior to Red Hat Linux 8.0, the default X Server configuration file was* /etc/X11/XF86Config-4*. The* -4 *was added because Red Hat once included two different major versions of the XFree86 Server, 3.a.b and 4.x.y. Since Red Hat has now dropped version 3.a.b, it has also dropped the* -4 *suffix. The XFree86 version 4.x.y server now includes data for all but the oldest graphics cards.*

Many of the directives in XF86Config are listed in Table 29.2. The following subsections correspond to the typical sections that you might see in your XF86Config file.

TABLE 29.2: COMMON DIRECTIVES IN */ETC/X11/XF86CONFIG*

Directive	Description
BoardName	Specifies the name assigned to the device, such as a video card.
BusID	Notes the location of a PCI or AGP video card, if Linux doesn't detect it.
DefaultDepth,Depth	Specifies the number of color bits per pixel; normally 1, 4, 8, 16, 24, or 32.
DisplaySize	Lists the horizontal and vertical size of the screen, in millimeters.
DRI	Specifies the Direct Rendering Interface.
Driver	Names a specific driver for the component.
EndSection	Indicates the end of a group of commands.
EndSubSection	Indicates the end of a SubSection group of commands.
FontPath	Notes where X fonts can be found; may cite a specific file, or the TCP/IP port of the local font server, usually with unix/:7100.
HorizSync	Shows the range of allowable horizontal synchronization rates for the monitor.
Identifier	Allows interaction between command groups.
InputDevice	May refer to keyboards or pointing devices such as a mouse or a touchpad.
Load	Adds the specified module.
Model Name	Represents the name of a specific model; goes with VendorName.
Mode	In the Monitor section, can detail monitor dot clock and timing. In the DRI section, this defines the permissions associated with the XF86 Server.

Continued on next page

TABLE 29.2: COMMON DIRECTIVES IN */ETC/X11/XF86CONFIG (continued)*

DIRECTIVE	DESCRIPTION
`Modes`	Specifies the allowable monitor resolution(s).
`Module`	Lists servers and font modules to be loaded with your X Server.
`Monitor`	Notes the monitor Identifier associated with a `Screen`.
`Option`	Indicates one of the many options available for different hardware components.
`RgbPath`	Notes a database file, in text format, that specifies the level of red, green, and blue for different colors.
`Screen`	Collects the information associated with the video card and displays and assigns available resolution modes.
`Section`	Indicates the beginning of a group of commands; should be labeled, and goes with `EndSection`.
`ServerLayout`	Collects the different components of the XFree86 server.
`SubSection`	Indicates the beginning of a group of commands inside a `Section`.
`VendorName`	Specifies the name of a manufacturer.
`VertRefresh`	Shows the range of allowable vertical refresh rates for the monitor.
`VideoRam`	Indicates the amount of available Video RAM memory.

SERVERLAYOUT

The `ServerLayout` section binds various `InputDevice`(s) and the `Screen`, which includes the combined configuration for the monitor and video card. The example shown here is in effect a summary of the configuration on my computer:

```
Section "ServerLayout"
     Identifier     "Default Layout"
     Screen    0    "Screen0" 0 0
     InputDevice    "Mouse0" "CorePointer"
     InputDevice    "Keyboard0" "CoreKeyboard"
     InputDevice    "DevInputMice" "AlwaysCore"
EndSection
```

In other words, this particular `ServerLayout` section combines the settings of `Screen0`, `Mouse0`, `DevInputMice`, and `Keyboard0`.

FILES

The `Files` needed by your X Server relate to colors and fonts. The example here is taken from my computer:

```
Section "Files"
     RgbPath    "/usr/X11R6/lib/X11/rgb"
```

```
    FontPath   "unix/:7100"
EndSection
```

To translate, this `Files` section notes the location of RGB style colors for display. It also lists the standard TCP/IP port for the X Font Server, `xfs`. RGB (Red Green Blue) is the traditional standard for color graphics.

Note *RGB is not good enough for many artists and graphic designers. There is an alternative. Some Linux applications support the CMYK (Cyan, Magenta, Yellow, and Black) standard. This is good enough for several major movie studios, including DreamWorks and Disney. A couple of Linux CMYK programs are Houdini (*`www.sidefx.com`*) and Maya (*`www.aliaswavefront.com`*).*

The X Font Server is critical to the X Window. If it's not running, you won't be able to start the X Window. In fact, if the X Font Server isn't running and your `/etc/inittab` file sets a default runlevel of 5, you'll need to boot your computer in single user or `linux rescue` mode in order to log into Linux.

Module

The `module` commands load font and server extension modules. The font modules are straightforward; they load the Freetype (True Type clone) and Type1 fonts. A full list of available modules is shown in the `/usr/X11R6/lib/modules` directory.

```
Section "Module"
    Load "dbe"
    Load "extmod"
    Load "fbdevhw"
    Load "glx"
    Load "record"
    Load "freetype"
    Load "type1"
    Load "dri"
EndSection
```

InputDevice

An `InputDevice` is anything that a user directly touches to send information to a computer. Also known as a Human Interface Device (HID), these devices are primarily keyboards and mice, but can include trackballs, touchpads, and more. As you can see below, there's a separate `InputDevice` section for each component.

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver     "keyboard"
    Option     "XkbRules" "xfree86"
    Option     "XkbModel" "pc105"
    Option     "XkbLayout" "us"
EndSection
```

This first InputDevice specifies your keyboard, using the driver by the same name. The basic keyboard rules specify a layout, which conforms to those associated with the XFree86 Server. The model is associated with a standard 105-key keyboard, in a standard U.S. layout.

```
Section "InputDevice"
     Identifier "Mouse0"
     Driver     "mouse"
     Option     "Protocol" "IMPS/2"
     Option     "Device" "/dev/psaux"
     Option     "ZAxisMapping" "4 5"
     Option     "Emulate3Buttons" "no"
EndSection
```

The next InputDevice specifies a mouse, using a PS/2 connection. The device driver file is /dev/psaux, which is often linked to /dev/mouse. ZAxisMapping represents the up and down motion of a mouse wheel, which in this case corresponds to standard mouse buttons 4 and 5. These buttons aren't available on all mice. Button 4 corresponds to a scroll wheel on a three-button mouse. Button 5 nominally corresponds to a button on the side of the mouse. If you have more than one mouse or pointing device, there may be another InputDevice section.

Monitor

The Monitor section summarizes the basic settings associated with your monitor. The following settings from my computer are fairly straightforward; they identify the monitor model, the DisplaySize in millimeters, and the horizontal sync and vertical refresh rates. The dpms option represents the power-saving settings standard.

```
Section "Monitor"
     Identifier  "Monitor0"
     VendorName  "Monitor Vendor"
     ModelName   "S/M 955DF"
     DisplaySize 360    270
     HorizSync   30.0 - 85.0
     VertRefresh 50.0 - 160.0
     Option      "dpms"
EndSection
```

It's possible to configure two different monitors; each monitor gets its own section with customized settings. The monitor and video card together gets its own Screen section, as we describe later in this chapter.

Device

The main device that supports any GUI is the video card. The following section identifies a specific card, with driver, and associated video RAM.

```
Section "Device"
     Identifier  "Videocard0"
     Driver      "i810"
```

```
        VendorName  "Videocard vendor"
        BoardName   "Intel 810"
        VideoRam    16384
EndSection
```

If you have more than one video card, each card gets its own separate section in your `XF86Config` file.

Screen

The `Screen` section combines the applicable video card (`Device`) and monitor settings from their respective sections. The name associated with the `Device` and `Monitor` lines is taken from their `Identifier` variables.

```
Section "Screen"
        Identifier "Screen0"
        Device     "Videocard0"
        Monitor    "Monitor0"
        DefaultDepth   24
        SubSection "Display"
              Depth    16
              Modes   "1024x768" "800x600" "640x480"
        EndSubSection
        SubSection "Display"
              Depth    24
              Modes   "800x600" "640x480"
        EndSubSection
EndSection
```

It's the combined video card and monitor that gets a dot pitch (`Depth`) and resolution (`Modes`). The following section configures two different `SubSection "Display"` stanzas. Note that each stanza has one `Depth` and possibly overlapping `Modes`.

DRI

The Direct Rendering Interface (DRI) takes advantage of the 3D acceleration available with higher-end video cards. It's associated with games as well as the higher-end graphics required for movies and computer-aided design models. The following `DRI` section is simple:

```
Section "DRI"
        Mode     0666
EndSection
```

The `0666` is associated with read and write file permissions, for all users. If you specify a group in `/etc/group`, you can limit 3D rendering access. For example, if there is a galley group in `/etc/group`, you could limit access with the following stanza:

```
Section "DRI"
        Group    "galley"
        Mode     0660
EndSection
```

Configuring Remote X Access

You don't have to run to a remote computer every time you need a GUI tool or application. The Linux GUI is built for networking. It's split into clients and servers. You can connect to an X server and display X client applications on local and remote computers.

This allows the computers of your choice to act functionally as application servers. In this section, we'll go through an example where the open source project management application, MrProject, is installed on one computer and can be opened on a second computer on that network.

Remote X access is disabled by default. You need to disable X security, preferably just for the computer clients of your choice. Getting the right commands on the correct computers can be confusing, so we've added a few explanations, which we've organized in Table 29.3.

- The X server is on the local computer. It is the computer where you want to run the GUI applications of your choice.
- The X client can be on a remote computer. For example, if you've installed Mozilla only on a remote computer, you can configure it to run on the local computer.
- Default security disables network access from remote X clients. You can enable it on the local X server with the correct `xhost` command.

Table 29.3: X Server and X Client configuration

X Server	X Client
Local computer, where you want to see the GUI clients.	Remote (or local) computer, where you run the commands to open the GUI clients.
Computer where you apply the xhost command, to allow X clients.	Computer where you allow Secure Shell access through any firewall.
To allow access from a remote PC named xclient, run `xhost +xclient`.	Secure shell access is simplest; otherwise, you'll configure xauth modules.

Allowing Access

To allow access to a networked X Server, there are two basic steps. First, you should set up the Secure Shell (SSH) on your X Client computers. It allows you to securely support access from remote computers. If you have a firewall, you'll want to customize it to support SSH access as described in Chapter 17. For more information on SSH, see Chapter 18.

Next, you'll want to go to the local computer to accept X commands from the remote computer. Let the remote computer in with the `xhost +`*`computername`* or `xhost +`*`remoteipaddr`* command. (*`remoteipaddr`* represents the IP address of the remote computer.) Then you can log into the remote computer and start a command such as `gimp` or `xclock`. As long as you've logged in through SSH, you don't even need to change the DISPLAY environment variable on the X Client.

Demonstrating a Remote Display

Now we'll show you how to set up a remote display, step by step. For the purpose of this exercise, assume that the local computer is named work, and the server with the MrProject application is named apps. Also assume that you have a user named michael on both computers.

1. We assume that MrProject is installed only on the X Client computer named apps and that you have a DNS server or an `/etc/hosts` file that correlates the names and IP addresses of each computer.
2. On apps, install the Secure Shell packages (if required), as described in Chapter 18. If apps has a firewall, make sure to customize it to allow SSH access.
3. Return to the computer named work. Log into the GUI, and open a command line terminal. Use SSH to log into apps remotely with a command such as:

   ```
   $ ssh michael@apps
   ```

4. Enter michael's password on computer apps when prompted. You should now be logged into the apps computer, where you can run MrProject with the `mrproject` command. The MrProject application should now appear on the local computer, work.

Troubleshooting the X Window

If you have problems starting the Linux GUI, there are a number of things that you can check. Much of this chapter has focused on the basic X configuration tools; you can always start by rerunning these tools.

As with most other servers, many problems can show up in the log files, stored in the `/var/log` directory. Sometimes the display is actually someplace else—on another console or even another computer. One common problem with starting the X Server is the fonts. If the X Font Server won't start, neither will the X Window.

Log Files

Two basic files are associated with events in the Linux X Window, and both are located in the `/var/log` directory. The `XFree86.0.log` file in this directory shows what happens when `startx` and associated commands interact with your configuration files, especially `XF86Config`. The `/var/log/messages` file can help you identify X Font Server problems.

Even if you're not having a problem, study these files. You may be surprised at the errors you find. What you learn can help you make your X Window start faster.

XFree86.0.log

Take a look at an excerpt from this log file in Figure 29.15.. If you've read the earlier section on the `XF86Config` file, you'll recognize many of the variables.

FIGURE 29.15
XFree86.0.log

```
Markers: (--) probed, (**) from config file, (==) default setting,
         (++) from command line, (!!) notice, (II) informational,
         (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/XFree86.0.log", Time: Mon May 24 16:52:32 2004
(==) Using config file: "/root/XF86Config"
(==) ServerLayout "Default Layout"
(**) |-->Screen "Screen0" (0)
(**) |   |-->Monitor "Monitor0"
(**) |   |-->Device "Videocard0"
(**) |-->Input Device "Mouse0"
(**) |-->Input Device "Keyboard0"
(**) Option "XkbRules" "xfree86"
(**) XKB: rules: "xfree86"
(**) Option "XkbModel" "pc105"
(**) XKB: model: "pc105"
(**) Option "XkbLayout" "us"
(**) XKB: layout: "us"
(==) Keyboard: CustomKeycode disabled
(**) |-->Input Device "DevInputMice"
(**) FontPath set to "unix/:7100"
(**) RgbPath set to "/usr/X11R6/lib/X11/rgb"
(==) ModulePath set to "/usr/X11R6/lib/modules"
(--) using VT number 7
                                                    20,8          3%
```

Make a note of those lines based on the configuration file, with the `"(**)"` in front. If there are problems, you can fix those in your `XF86Config` file. In my version of the file, I see lines such as:

```
(II) I810(0): Not using default mode "320x175" (bad mode
➥clock/interlace/doublescan)
```

This is an informational `(II)` message, since it doesn't affect how things work. But look for warning `(WW)` and error `(EE)` messages.

LEARNING TO TROUBLESHOOT

Troubleshooting can be a difficult process. You can wait until trouble strikes; crises do have a tendency to focus the mind. Alternatively, you can experiment. Because the X Window depends on the `XF86Config` file, I learn about possible problems by experimenting on this file. If you know the `linux rescue` mode described in Chapter 11 and are systematic, you too can learn this way.

Before experimenting with any configuration file, back it up. In this case, make sure your `id` variable in `/etc/inittab` is set to runlevel 3. If you run into problems with `XF86Config`, this will help you restart Linux at the command-line interface.

Try "commenting out" various commands in this file, by adding an "#" in front of the line, and then run `startx`. Sometimes your X Window will start fine, using other settings as defaults. Other times, your X Window might not start at all. Pay attention to the `(EE)` messages and their relationship to what you changed in the `XF86Config` file.

When you've finished, remember to restore the original configuration file.

/VAR/LOG/MESSAGES

The X Window can't start unless your X Font Server is running. It's a service controlled from the `/etc/rc.d/init.d` directory, like many other services.

NOTE *The main font configuration file is* `/etc/fonts/fonts.conf`.

The `/var/log/messages` file is fairly long. By default, it can hold the startup and shutdown messages for your Linux computer for up to a full week. If the problem is recent, start near the end of the file. The first message you'll see during the startup process should look like this:

```
Dec 22 10:25:09 Enterprise3 kernel: Linux version 2.4.21-12.EL
```

This will be followed up by an `xfs` startup message similar to the following:

```
Dec 22 10:25:09 Enterprise3 xfs: xfs startup succeeded
```

If you don't see this message, you may have a font problem. Look at the following possibilities:

- Check the status of the `xfs` service. If it's stopped, try starting it with the `service xfs start` command. Make sure `xfs` is set to start automatically with the appropriate `chkconfig` command, discussed in Chapter 13.
- Check the `FontPath` variable in `/etc/X11/XF86Config`. It should point to actual font files or TCP/IP port 7100.
- Make sure the files listed in the `FontPath` variable actually exist. If they don't, you may need to install some of the font RPM packages associated with XFree86. These packages have names in a format like `XFree86-*-fonts-*`.
- Check your firewall. If you're blocking local access to port 7100, the font server can't get information to your X Window.

NOTE *Don't confuse the X Font Server with the xfs file system developed by Silicon Graphics (SGI). Unfortunately, they do use the same acronym.*

Summary

In this chapter, you learned the basics of configuring the X Window. While many Linux experts have no desire or need for the graphical user interface, it is an important tool for many power users. It holds appeal for users who are converting from more graphical systems, such as Microsoft Windows.

You may have already configured the X Window during the Linux installation process. If you haven't or need to change your settings, you can use the Red Hat Display Tool, which you can start with the `redhat-config-xfree86 command`. The alternative `xf86config` is available on other distributions or if you download new XFree86 servers from `www.xfree86.org`.

Several key configuration files are associated with the X Window, called through the `startx` script. You can create individual settings in your home directory, or allow `startx` to use generic settings in the `/etc/X11` directory.

Linux is so client-server focused that it even allows you to configure the X Window into clients and servers. The X server is on the local computer; you can call GUI applications, also known as X clients, from remote computers. With some simple `xhost` commands, you can then use SSH to start remote X clients.

Perhaps the key configuration file is `/etc/X11/XF86Config`. It's helpful to know the basics of this file, so you can customize it as well as troubleshoot some of the problems you may encounter. While the X Window requires a working font server, `xfs`, you'll find most problems in the main X Window log file, `/var/log/XFree86.0.log`.

In the next chapter, we'll take a detailed look at the default desktop for Red Hat Enterprise Linux, GNOME. It is a fully featured GUI, with virtually all of the features available on Microsoft Windows. Even if you don't use a GUI, you should know the benefits of GNOME in order to help your users.

Chapter 30

The Red Hat GUI Workstation

WHILE LINUX ADMINISTRATORS MAY not need a graphical user interface (GUI), users who are converting from Microsoft Windows do. One of the goals within the GNU community is to make the Linux operating system competitive on the desktop. And progress is being made, as shown by the *Wall Street Journal* on May 24, 2004: "Can Linux Take Over the Desktop?" To this end, Linux needs a GUI that can help Microsoft Windows users feel comfortable.

As described in Chapter 29, there are two major desktop environments: GNOME and KDE. To make progress on the desktop, Red Hat has integrated the Bluecurve theme into its implementations of both GNOME and KDE. It also has integrated a number of common tools into the main menus of both desktops. In Red Hat Enterprise Linux, the two desktops are converging in functionality, courtesy of the Red Hat Bluecurve theme. Thus, your choice of desktop is a matter of personal preference. Not only does this provide a high-performance GUI, but it also includes high-performance software such as office suites that can cost the Microsoft user hundreds of dollars. As noted in Chapter 1, this has caused a number of companies and governments to consider replacing Microsoft Windows with Linux.

KDE and GNOME provide a GUI desktop, control applets, and several important applications. Many of these components can replace costly third-party applications that run only on Microsoft Windows. In this chapter, we'll briefly cover the Red Hat desktop interfaces, basic applications, office suites, graphical applications, and more.

However, this chapter cannot provide a comprehensive introduction to the Linux desktop. It also provides a brief overview of Linux office suites and what you can do to customize a desktop for different languages. For more information on the GNOME desktop, see *The Official GNOME 2 Developer's Guide* by Matthias Warkus. This chapter covers the following topics:

- Working with the basic GNOME and KDE interfaces
- Customizing a workstation
- Learning about common GNOME and KDE extras
- Touring the OpenOffice.org suite
- Opening graphical applications
- Setting default languages

Working with the Basic GNOME and KDE Interfaces

The standard Red Hat GNOME and KDE desktops have all the characteristics of today's GUI operating systems. Each desktop includes a panel, a Main Menu button, and icons. You can customize each of these components for your own needs or even configure a standard interface. You can control and customize the look and feel through the GNOME or KDE Control Centers. When you first start a GUI on Red Hat Enterprise Linux, you'll probably see a desktop similar to Figure 30.1 or 30.2. The Red Hat defaults for both desktop interfaces include the Bluecurve theme, which makes these desktops look as common as possible.

NOTE *In Red Hat Enterprise Linux, the panel is functionally equivalent to the Microsoft Windows taskbar; the Main Menu button (with the red hat icon) corresponds functionally to the Microsoft Start button.*

FIGURE 30.1
The Red Hat GNOME desktop

The Desktop, as Homogenized by Red Hat

The basic GNOME desktop is deceptively simple. As you can see from the first two figures, it includes a way to navigate to your home directory (*username*'s home); a Start Here button that opens available applets, utilities, and applications; and a Trash folder. All three use an Explorer-style graphical shell that you use to manage your files, your GNOME configuration, and any GUI tools associated with your Linux system. We'll use Nautilus later in this chapter. In the following sections, we'll examine the buttons on the panel and the GNOME Control Center.

FIGURE 30.2
The Red Hat KDE desktop

THE PANEL

The GNOME and KDE panels allow you to call up a number of applications, switch between open programs, and even switch between open workspaces. I've included a view of my GNOME and KDE panels in Figures 30.3 and 30.4. This may not include all the icons that you see on your desktop. In this case, the GNOME panel includes seven icons on the left, which are briefly described in Table 30.1.

NOTE *A workspace is like a standard desktop, with its own icons and open programs. By default, GNOME includes four workspaces; the data for three are stored in spare video memory.*

FIGURE 30.3
The GNOME panel

FIGURE 30.4
The KDE panel

TABLE 30.1: PANEL ICONS

ICON	DESCRIPTION
	Clicking this button opens the Main Menu, which provides access to available programs and utilities; it works like the Start button in Microsoft Windows. Since this is trademarked by Red Hat, the button is different if you're using a rebuild distribution.
	Opens the Mozilla web browser.
	Starts the Evolution personal information manager; functionally similar to Microsoft Outlook.
	Begins the OpenOffice.org Writer; functionally similar to Microsoft Word.
	Opens the OpenOffice.org Impress presentation manager; similar to Microsoft PowerPoint.
	Starts the OpenOffice.org Calc spreadsheet program; similar to Microsoft Excel.
	Begins the GNOME Print Manager.

THE MAIN MENU

Now we'll take a brief look at the Main Menu. Click the red hat in the lower-left corner of the desktop. You should see a menu style that should look familiar if you've used other Linux or Microsoft Windows desktops.

The Main Menu opens a series of other commands and menus. They vary slightly depending on whether you've using the GNOME or KDE desktop. They are briefly described in Table 30.2.

TABLE 30.2: MAIN MENU COMMANDS AND SUBMENUS

MENU OR COMMAND	DESCRIPTION
Accessories	Opens a group of small programs, such as text editors and calculators.
Documentation	Enters any documents that you may have loaded from the Red Hat Enterprise Linux documents CD.
Games	Navigates to any games that you may have installed.
Graphics	Accesses graphics applications for editing, screenshots, faxes, PDF readers, and more.
Internet	Includes a series of applications that you can use to communicate on a TCP/IP network such as the Internet.

Continued on next page

TABLE 30.2: MAIN MENU COMMANDS AND SUBMENUS *(continued)*

Menu or Command	Description
Office	Opens a group of applications associated with the OpenOffice.org suite of programs; other office suites should be accessible through this menu.
Preferences	Allows you to customize your settings; mostly related to the desktop.
Programming	Opens access to a group of programming tools; strangely enough, Emacs is part of this group.
Sound & Video	Adds multimedia applications, including a CD writer.
System Settings	Includes access to many `redhat-config-*` administrative utilities; most require root-level access.
System Tools	Starts a menu with a variety of administrative tools.
Control Center	The KDE desktop links to the KDE Control Center from the Main Menu.
Help	Opens a help session in a simplified browser.
Home Folder	Starts a browser with a view of the files in your home directory.
Network Servers	Used in GNOME only; provides access to shared folders from other computers, including Microsoft Windows computers via Samba.
Run Program (KDE Run Command)	Opens a Run Program dialog box where you can type in the text name for an application.
Search For Files (KDE Find Files)	Starts a front end to the Find command for file searches, starting from a specified directory.
Open Recent	Used in GNOME only; allows you to open recently accessed documents; normally uses OpenOffice.org.
Lock Screen	Starts a secure screensaver; to return to the desktop, you need your username and password.
Log Out	Exits the GUI.

If you don't see a specific menu, you may not have installed the associated package(s). For example, you won't see a Games menu unless you've installed associated packages such as `gnome-games-*`.

The Control Centers

Now we'll take a very brief look at some more detailed configuration options for each desktop, associated with each Control Center. In the GNOME desktop, you can open the GNOME Control Center from the Main Menu: click Main Menu ➢ Preferences ➢ Control Center. This opens a Nautilus window with a series of icons, shown in Figure 30.5. Every icon is associated with a graphical application that can help you work with GNOME. The Additional Preferences icon opens a different Preferences window. We describe each of the applets in Table 30.3.

FIGURE 30.5
The GNOME Control Center

NOTE *The applets in the GNOME Control Center are also available in the Main Menu ➢ Preferences submenu.*

TABLE 30.3: GNOME CONTROL CENTER OPTIONS

OPTION	DESCRIPTION
More Preferences	Acts as a gateway to additional Control Center applets, including the CD Database, desktop switcher, file management, Palm connection, Panel, and Sessions
About Myself	Starts a front end to the `chfn` command for user information
Accessibility	Opens an interface to modify keyboard behavior
Background	Supports changes to the GUI desktop background
CD Properties	Configures automount and data preferences for CDs and DVDs
Control Center	Opens a second Preferences window with the same Control Center applets
File Types And Programs	Allows you to associate file types and applications
Font	Starts a Font Preferences customization window
Keyboard	Defines keyboard repeat and cursor blink speed
Keyboard Shortcuts	Associates keyboard combinations with different functions
Login Photo	Configures a background in the `gdm` login window
Menus & Toolbars	Allows you to modify the look and feel of icons and text in program menus
Mouse	Manages the behavior of mouse actions and cursors

Continued on next page

TABLE 30.3: GNOME CONTROL CENTER OPTIONS *(continued)*

OPTION	**DESCRIPTION**
Network Proxy	Configures a path for an external network connection
Password	Changes the login password
Preferred Applications	Allows you to assign a preferred web browser, text editor, and command-line terminal
Screensaver	Configures a specific or random screensaver from a list
Sound	Associates sound files with specific events
Theme	Lets you configure a theme for the desktop environment; the default is Bluecurve
Windows	Configures the behavior of windows to certain actions

If you're in the KDE desktop, you have access to a fairly comprehensive configuration tool, the KDE Control Center. It allows you to configure the desktop—and a lot more. You can open it from the Main Menu: click Main Menu ➢ Control Center. This opens the Control Center window, shown in Figure 30.6. As you can see, the Control Center allows you to configure your computer in a number of areas, from Appearance & Themes to Web Browsing. Each tool in these areas will be covered in the following sections.

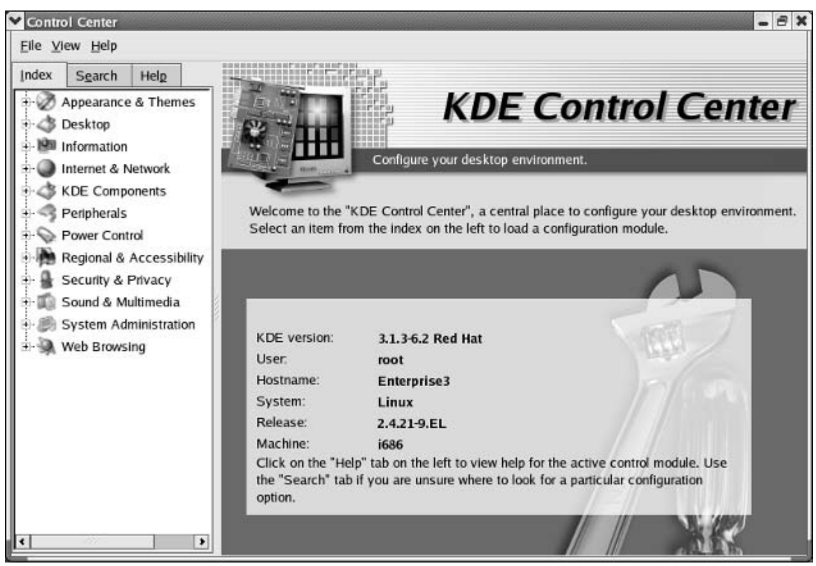


FIGURE 30.6
The KDE Control Center

Whenever you make a configuration change, you should click the Apply button to write the changes to your `~/.kde` directory. (As described in Chapter 8, the ~ represents your home directory.) We describe each of the main menus in Table 30.4.

NOTE *Some applets in the KDE Control Center require administrative access. If you're working as a regular user (not root), you'll see an Administrator Mode button when required. If you want to make changes to these types of settings, click the Administrator Mode button and enter the root password in the Run As Root window that appears.*

TABLE 30.4: KDE CONTROL CENTER MENUS

MENU	DESCRIPTION
Appearance and Themes	Allows you to customize the look and feel of your KDE desktop
Desktop	Includes more options customizing the look and feel of your KDE desktop
Information	Provides a graphical view of detected hardware system information, mostly from `/proc`
Internet & Network	Supports configuration of parameters for network and shared directories
KDE Components	Allows configuration of basic parameters for some KDE utilities, such as the address book and log out defaults
Peripherals	Provides an interface for four external devices (if installed): a digital camera, a keyboard, a mouse or other pointing device, and a printer
Power Control	Allows you to configure power management settings, including those associated with laptop batteries, if installed
Regional & Accessibility	Supports formats associated with various nations and languages; sets up keyboards with bells.
Security & Privacy	Configures encryption and password settings.
Sound & Multimedia	Lets you configure settings associated with your sound card, music files, and more.
System Administration	Enables you to configure a variety of administrative settings
Web Browsing	Allows you to configure your KDE web browsing experience, through the default KDE web browser, Konqueror

Customizing a Workstation

When you configure workstations for users, you're normally expected to customize these workstations. In some cases, departments expect a common "look and feel"; in other cases, you may be customizing settings for an executive. Modifying the basic look of the desktop is elementary.

You can customize the programs that start with your GUI from the command-line interface using the `.xinitrc` file discussed in Chapter 29. Just make sure to point the last command to the appropriate desktop environment, using one of the following commands:

```
exec gnome-session
```

```
exec startkde
exec twm
```

Alternatively, you can use one of the session management tools described in the following sections. Both GNOME and KDE include their own tools that your users can run to customize their own desktop environments.

We recommend you use either `.xinitrc` or one of these GUI tools, not both. Otherwise, the effect of these tools are cumulative, and configuration control may be difficult.

GNOME Customization

In GNOME, you can customize the look of the desktop using various Control Center applets that we previously described. Important applets in this area include Background, Menus & Toolbars, Screensaver, and Theme.

The GNOME Sessions tool allows users to customize their own desktop environments. It's fairly easy to use. You can start it from the Main Menu: click Main Menu ➤ Preferences ➤ More Preferences ➤ Sessions.

The Sessions window allows you to configure the programs that start when you enter the GNOME desktop. It also allows you to configure the behavior when GNOME starts and monitors currently loaded programs. As shown in Figure 30.7, the window contains three tabs, described in Table 30.5.

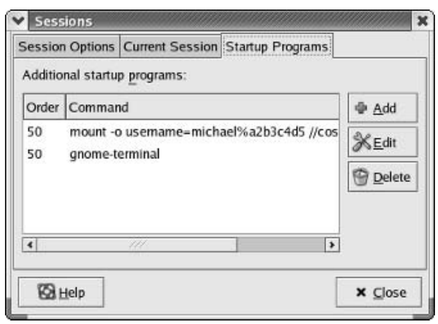


FIGURE 30.7 GNOME Sessions management

TABLE 30.5: GNOME SESSIONS

TAB	DESCRIPTION
Session Options	Manages behavior during the GNOME login and logout process
Current Session	Lists currently running programs in GNOME
Startup Programs	Notes the programs that start when GNOME starts

The Startup Programs tab is key. You can add the programs of your choice, such as those shown in Figure 30.7. Click Add, and enter the text command of your choice. The PATH associated with

your account applies; for example, if you want to set a GNOME terminal session to start when you start the GNOME desktop, you don't need to enter the full path to the `gnome-terminal` command.

KDE Customization

In KDE, you can customize the look of the desktop through the KDE Control Center Appearance & Themes and Desktop Menus. You can configure desktop wallpaper, screensavers, and more under these submenus.

KDE by default preserves the programs and applications that you open. If you close the KDE desktop environment without closing these programs, KDE will open these programs the next time you restart this GUI. You can modify the associated settings using the Control Center, the KDE Components menu, or the Session Manager submenu.

Learning Common GNOME and KDE Extras

Both GNOME and KDE come with a number of bonus applications, accessible through the Main Menu button. They fall into several categories: Accessories, Internet, Preferences, Multimedia, and System Tools. This is not a comprehensive list of programs available through the Main Menu button. While most are based on the work of the GNOME and KDE projects, a few third-party utilities are included in this chapter as well.

If you don't see a particular application or extra in your own GNOME or KDE desktop environment, you may not have installed the associated `rpm` package. As this is not a comprehensive guide to a Linux GUI desktop, we have not included every tool that you can install on a Red Hat Enterprise Linux 3 workstation.

In some cases, updates are required for specific applications. Some instant messengers may have upgraded their software, and some browsers may not work without the latest Java client. If an update isn't available from Red Hat, you may be able to download an update from the developers of the application. Remember, Red Hat Enterprise Linux is a distribution, which means that most of the software in this operating system was developed by a number of third parties.

Accessories

GNOME includes several accessories that help you with simple computing tasks. They're accessible through the Main Menu ➢ Accessories submenuThere are slight variations between the GNOME and KDE desktops in this submenu. For a brief overview of these extras, see Table 30.6. GNOME and KDE locate some of these extras in the main Accessories submenu or the More Accessories submenu. In many cases, the GNOME and KDE versions of each utility are subtly different (or more).

TABLE 30.6: ACCESSORIES MENU

OPTION	DESCRIPTION
Calculator	Starts a scientific calculator that connects to a standard keyboard numeric keypad.
Character Map	Opens an interface with characters associated with Roman-style alphabets

Continued on next page

Table 30.6: Accessories Menu *(continued)*

Option	Description
Dictionary	Connects to an online dictionary server at `dict.org`.
File Roller	Provides a front-end similar to Windows Zip utilities for `tar.gz` Tarballs.
Handheld PDA	Opens a Pilot Link utility.
Print Manager	Starts the GNOME print manager.
Text Editor	Opens a GUI text editor.
Address Manager	Starts a contact information manager.
KAlarm	Goes to the KDE-based event-based alarm utility
Kandy	Supports syncing between an address book and a mobile phone.
KArm	Helps you track the time that you spend on different tasks.
Kdeprintfax	Allows you to view a file that you've printed to a fax device.
KHexEdit	The KHexEdit utility is a customizable hex editor that can display and help you edit data in hexadecimal, octal, and binary modes. (That corresponds to base 16, base 8, and base 2 for the math majors.) It can also show files in text mode.
KJots	The KJots utility lets you jot down short notes in an organized fashion. Any "books" you create can be added to a hotlist.
KNotes	The KNotes utility allows you to add some short notes to a list that you can print or e-mail.
KOrganizer	The KOrganizer is a handy scheduling utility.
KPilot	The KPilot utility uses the latest version of Desktop HotSync software; it is intended as a substitute for Palm desktop software.
KTimer	KTimer allows you to start a command after a given delay; the default is 100 seconds.

Documentation

If you've installed the packages from one of the Red Hat Enterprise Linux documentation CDs, you'll see a Documentation menu, which supports easy access to these documents in the default Web browser. This is just for your convenience; PDF versions of Red Hat Enterprise Linux manuals are available on the documentation CD.

Games

If you've installed any standard GNOME or KDE games packages on your system, you'll be able to access them through Main Menu ➢ Games. We do not cover the startup or operation of any Linux games. On the other hand, there are some who believe that games can give Microsoft Windows users more comfort during any transition to Linux.

Internet Utilities

GNOME includes a number of utilities and applications for communicating on the Internet. The difference between a utility and an application in this case is somewhat arbitrary; we'll look at the Mozilla browser, the Ximian Evolution personal information manager, and the Gaim instant messaging (IM) utility from the Internet Applications section.

In this section, we'll take a brief look at more basic programs, including instant messengers, chat programs, and other miscellaneous connection utilities. These programs are available through Main Menu ➢ Internet ➢ More Internet Applications. For a brief description of these programs, see Table 30.7. Some are associated with GNOME; others with KDE.

TABLE 30.7: ACCESSORIES MENU

OPTION	DESCRIPTION
Ethereal	Opens a protocol analyzer; see Chapter 17.
gFTP	Starts a graphical FTP client; see Chapter 22.
IRC Client	Accesses an Internet Relay Chat client.
KGet	Goes to a KDE download manager.
Kit	Opens a KDE Instant Messaging client for AOL.
KMail	Starts a KDE email client.
KNewsTicker	Accesses a news ticker; the default configuration includes links to Slashdot stories.
KNode	Goes to a KDE Newsgroup reader
Konqueror	Opens the KDE Web browser; relatively "light" compared to Mozilla.
Korn	Starts an incoming email monitor.
KPPP	Accesses the KDE PPP connection utility, which supports connections to ISPs over a telephone modem.
KSirc	Goes to a KDE IRC chat client.
Mozilla Mail	Opens the Mozilla email client
Mozilla Mail Message	Starts the Mozilla email client, configured for an outgoing message.
Remote Desktop Connection	Supports access to a VNC server on a remote computer.

Internet Applications

In this section, we'll cover three basic applications commonly run by Linux users on the Internet. These are Red Hat default programs: the web browser, Mozilla; the Personal Information Manager, Evolution; and the Instant Messenger (IM) client, Gaim, which are accessible via Main Menu ➢ Internet.

Mozilla

The default Red Hat web browser is Mozilla. It is a fully featured web browser, built on the code that Netscape re-released as open source in 1998. You can navigate between several features by clicking on the icons shown in the lower-left corner of the window (see Figure 30.08). From left to right, these icons are associated with a web browser, mail and newsgroups reader, web page composer, address book, and IRC chat client.

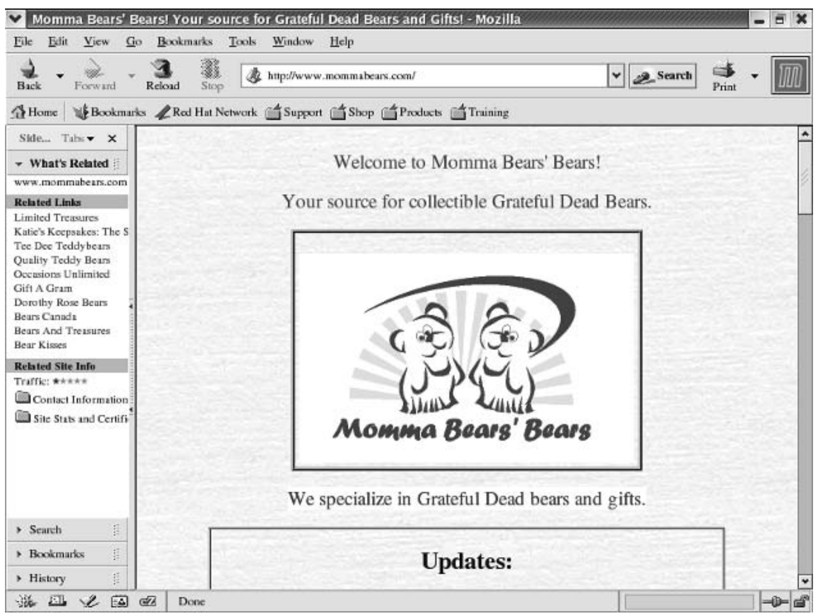
Figure 30.8
Mozilla icons

Note *When Marc Andreesen was working on the Netscape Web browser, the leading browser was known as Mosaic, and he wanted a "Mosaic Godzilla," which became the code name for the browser project: Mozilla.*

The Mozilla Browser

The default Mozilla web browser has the same look and feel as Netscape, as shown in Figure 30.9. It includes commands associated with Netscape, such as the What's Related sidebar.

Figure 30.9
Mozilla web browser



You can customize various Mozilla settings. Click Edit ➢ Preferences to open the Preferences window. The options and wizards should be familiar to Netscape users.

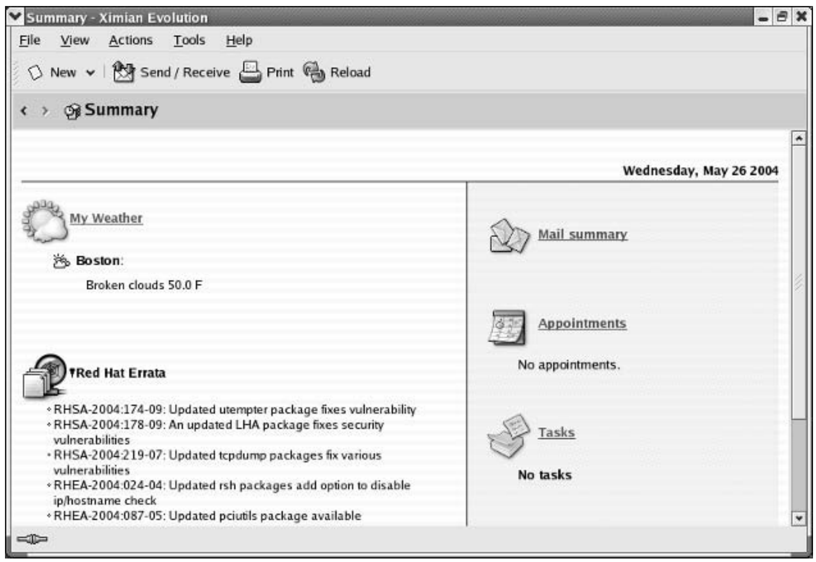
Evolution

While the name of the program on the GNOME desktop is Evolution Email, it is so much more. It serves as a personal information manager, similar to Microsoft Outlook.

When you first start Evolution, you're prompted to configure your profile. Evolution can handle all types of standard e-mail, including POP, IMAP, and MH-style servers. It also requires that you identify your time zone, and it prompts you to import your address book and e-mail from other formats.

As you can see in Figure 30.10, the Evolution summary view lists the status of your local weather, e-mail, appointments, and upcoming tasks. One additional useful feature for Linux administrators is a list of the latest Red Hat errata.

Figure 30.10
Ximian Evolution



Note *Ximian (*`www.ximian.com`*) is an important player on the Linux desktop, developing GUI desktop tools for the Enterprise. They've It has also launched the Mono project, which is working toward an open source implementation of Microsoft's .NET platform.*

Instant Messenger

GNOME includes an instant messenger (IM) client, suitable for connections to a variety of servers, including those provided by America Online (AOL), Yahoo!, the Microsoft Network (MSN), and more. The official acronym is Gaim, which is short for a GNU version of some popular IM program (which I should not name). In reality, the acronym does not do Gaim justice, as unlike the proprietary IM programs, Gaim uses plug-ins, which are essentially program adapters, to connect to several different types of IM networks. The Gaim login screen is shown in Figure 30.11.

FIGURE 30.11
Gaim login screen

To access a specific network, click Accounts. You can then select and configure an account on the network of your choice.

For details on required login information, consult the IM provider of your choice. The Screen Name corresponds to your account; the Alias is what is typically seen in the IM chat area. I've used Gaim on the Microsoft Network, and as of this writing, it includes additional useful emoticons.

Preferences

Most of the utilities associated with the Main Menu ➢ Preferences submenu were covered earlier in this chapter, in the discussion on the GNOME Control Center. This section deals with the utilities associated with the Main Menu ➢ Preferences ➢ More references submenu.

Multimedia

Several GNOME multimedia applications are available when you click Main Menu ➢ Sound & Video. These include various audio and CD players and sound control utilities. For a brief overview of these utilities, see Table 30.8. The menu is subtly different between GNOME and KDE. I do not cover all of the utilities that you might see in both desktop environments.

TABLE 30.8: MULTIMEDIA MENU

PROGRAM	DESCRIPTION
Audio Player	Starts the X Multimedia System (XMMS).
CD Player	Opens a Audio CD player; the KDE version is KsCD.
Sound Recorder	Goes a utility where you can record sounds, play .wav files, and mix different sounds.
Volume Control	Accesses the control for volume for a number of systems.
Grip	Starts the GNOME CD player and burner.
aRts Builder	Opens a sound server.

Continued on next page

Table 30.8: Multimedia Menu *(continued)*

Program	Description
Kaboodle	Goes to a media player for single files.
KMid	Supports midi (`.mid`) and karaoke (`.kar`) file formats.
KMidi	Opens a front end to the Midi synthesizer

System Settings

There are a wide variety of system settings programs available. Many are `redhat-config-*` tools which covered in other chapters. For a brief overview of these utilities, see Table 30.9.

Note *If you're familiar with Fedora Linux, you'll find these* `redhat-config-*` *tools with a slightly different name:* `system-config-*`.

Table 30.9: System Settings

Program	Description
Desktop Switching Tool	Opens the `switchdesk` tool, which allows you to set the default GUI desktop environment.
Domain Name System	Starts the Domain Name Service server management tool (`redhat-config-bind`)
HTTP	Goes to the HTTP Web server configuration tool (`redhat-config-httpd`).
Network Booting Service	Accesses the Network Installation and Diskless Environment tool, which supports diskless workstations and network installations (`redhat-config-netboot`).
NFS	Opens the NFS Server Configuration tool (`redhat-config-nfs`)
Samba	Starts the Samba Server Configuration tool (`redhat-config-samba`)
Services	Goes to the Service Configuration tool (`redhat-config-services`)
Add/Remove Applications	Accesses the Red Hat Package Management tool (`redhat-config-packages`)
Authentication	Opens the Red Hat Authentication tool (`redhat-config-authentication`)
Date & Time	Starts the Date/Time Properties tool (`redhat-config-date`).
Display	Goes to the Display Settings tool (`redhat-config-xfree86`).
Keyboard	Accesses the Keyboard tool (`redhat-config-keyboard`).
Language	Opens the Language Selection tool (`redhat-config-languages`).
Login Screen	Starts the GDM Setup tool (`gdmsetup`).
Mouse	Goes to the Mouse Configuration tool (`redhat-config-mouse`).
Network	Accesses the Network Configuration tool (`redhat-config-network`).

Continued on next page

TABLE 30.9: SYSTEM SETTINGS *(continued)*

PROGRAM	**DESCRIPTION**
Printing	Opens the Printer Configuration tool (`redhat-config-printer`).
Root Password	Allows the root user to change the root password (`redhat-config-rootpassword`).
Security Level	Starts the Security Level Configuration tool (`redhat-config-securitylevel`).
Soundcard Detection	Opens the Audio Devices tool, which tells you if Red Hat has detected a sound card on your computer (`redhat-config-soundcard`).
Users and Groups	Accesses the Red Hat User Manager (`redhat-config-users`).

System Tools

There are a wide variety of system tools available. For a brief overview some of these tools, see Table 30.10.

TABLE 30.10: SYSTEM TOOLS

PROGRAM	**DESCRIPTION**
Disk Management	Starts the User Mount tool, which illustrates the current state of mounted Linux filesystems, based on `/etc/fstab`. Can be used to mount or format a filesystem.
Floppy Formatter	GNOME only; opens a utility to format floppy drives to Linux or DOS formats.
Hardware Browser	Illustrates detected devices on your computer, for information only.
Info Center	KDE only; connects to the KDE Control Center Information menu.
Internet Configuration Wizard	Navigates to the Red Hat Internet configuration druid, `redhat-config-network-druid`; see Chapter 16 for more information.
Kickstart	Opens the Red Hat Kickstart Configurator; see Chapter 5 for more information.
Network Device Control	Allows you to activate or open the Network Configuration tool for configured devices.
Printing Notification Icon	Activates drag-and-drop printing.
Red Hat Network	Sets defaults for your `up2date` connection to the Red Hat network management servers.
Red Hat Network Alert Icon	Adds a circular icon to your taskbar; should already be installed by default.
Screen Resize and Rotate	KDE only; Starts a taskbar icon which makes it easier to resize your display.
System Logs	Opens the Red Hat System Log tool, `redhat-logviewer`.
System Monitor	Monitors current processes; CPU and swap partition usage.

Continued on next page

Table 30.10: System Tools *(continued)*

Program	**Description**
Terminal	Starts the standard command-line interface for the desktop environment (`gnome-terminal` or `konsole`).
Configuration Editor	Navigates to GConf, which is a front-end to the settings stored in users' home directories.
Desktop Sharing	Allows you to invite others to connect to a local configured VNC server.
File Manager	KDE only; opens Konqueror as a File Manager;
KAudioCreator	Starts a multimedia application to copy tracks from audio CDs for writing to the CDs of your choice.
KDE System Guard	Navigates to the KDE System Load tool, a front end to the `top` command.
KDiskFree	Opens a KDE front-end to the `df` command.
Kernel Tuning	Starts the Red Hat Kernel Tuning tool, `redhat-config-proc`.
KRec	Goes to a recording fron end to the KDE aRts sound server.
Mail Transport Agent Switcher	Opens the `redhat-switch-mail` utility, which allows you to switch between installed mail servers, namely Postfix and sendmail.

Touring the OpenOffice.org Suite

Perhaps the biggest bonuses with Red Hat Enterprise Linux are the fully featured office suites. These are programs that you can substitute for Microsoft Office that can cost hundreds of dollars per computer. All are interchangeable to some degree with Microsoft Office; in fact, some Linux-based office suites can now handle the macros that have made it difficult to use Microsoft Office-based documents on other systems.

The Linux office suites typically include a word processor, spreadsheet, graphics support, presentation manager, and a project scheduler. Some suites include more. You might need to download an application or two, but they are as freely available as the office suite applications that come with Linux. The Office suite included with Red Hat Enterprise Linux 3 is OpenOffice.org.

One thing we'll review in some detail are compatible file formats, so you can make some basic judgments about using a Linux office suite as a replacement for something like Microsoft Office. However, the details depend on the data in your files; for example, OpenOffice.org Calc may not be able to handle every macro. Once you test your data, you and your users can have some degree of confidence that you can replace a Microsoft office with one of these freely available suites.

OpenOffice.org was developed from the same code as Sun Microsystems' StarOffice. It includes several applications, which are briefly described in Table 30.11.

You can open installed OpenOffice.org applications from the GUI of your choice. Click Main Menu ➢ Office and then select the application of your choice from the menu that appears.

Alternatively, you can start three OpenOffice.org applications directly from the panel at the bottom of the desktop. As we described earlier in this chapter, you can start OpenOffice.org Writer by clicking

TABLE 30.11: OPENOFFICE.ORG APPLICATIONS

APPLICATION	DESCRIPTION
Calc	Spreadsheet
Draw	Diagram creator
Impress	Presentation manager
Math	Formula creator
Printer Setup	Administers a printer interface
Writer	Word Processor

on the icon of a pen and paper; you can start OpenOffice.org Impress by clicking on the icon of a bar graph and slide; and finally, you can start OpenOffice.org Calc by clicking the icon of a graph pie chart.

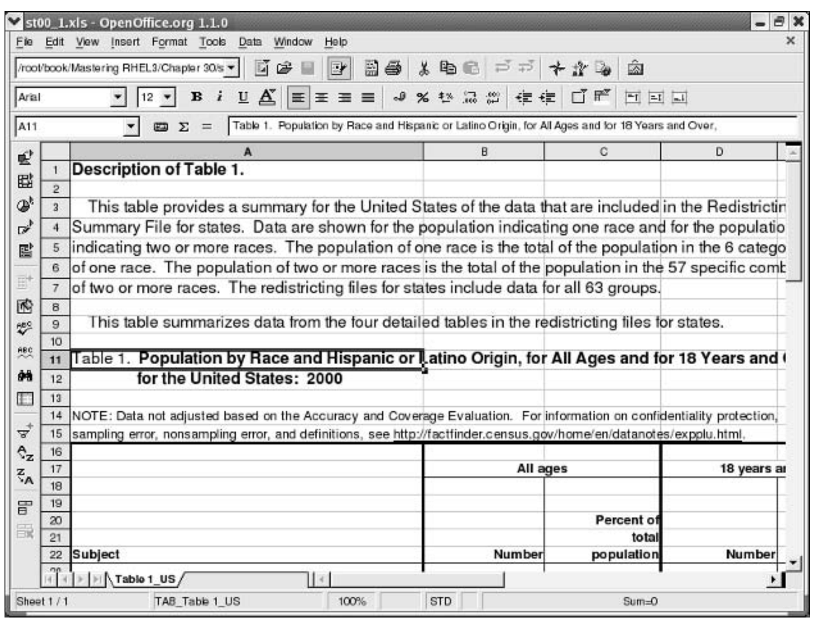
You can learn more about the OpenOffice.org project from their web site.

OpenOffice.org Calc

Perhaps the first key business PC application was the spreadsheet. You can use a spreadsheet to define a range of numbers. With the equations of your choice, you can set up a spreadsheet to perform a variety of calculations in different scenarios. It's useful for everything from statistical analysis to business modeling and projections.

You can open OpenOffice.org Calc by selecting Main Menu ➤ Office ➤ OpenOffice.org Calc, or by running the `oocalc` command from a GUI terminal window. Figure 30.12 shows Open-Office.org Calc and some basic data from the year 2000 U.S. census.

FIGURE 30.12
OpenOffice.org Calc and the census



As you can see in the figure, OpenOffice.org Calc has the basic look and feel of a spreadsheet. Functionally similar to Microsoft Excel, Calc includes several toolbars, as described in Table 30.12.

TABLE 30.12: OPENOFFICE.ORG CALC TOOLBARS

Toolbar	Description
Formula	Reflects the cell location and any formulas associated with that cell
Function	Configures basic functions such as open, print, and undo
Hyperlink	Sets up access to web pages
Main	Allows the creation of charts with graphs; supports format and spell checks; permits sorting and grouping
Object	Supports formatting options, including fonts, justification, numbering systems, borders, and alignment

OpenOffice.org Calc works with many different types of spreadsheets, including the formats described in Table 30.13. As you can see, OpenOffice.org Calc can work with spreadsheets from a number of applications, including Microsoft Excel, StarOffice Calc, dBASE/FoxPro databases, and more. You can also set up OpenOffice.org Calc with text files in comma-separated format (see the accompanying sidebar).

TABLE 30.13: OPENOFFICE.ORG CALC FILE FORMATS

Format	Description
`.sxc`	OpenOffice.org Spreadsheet
`.stc`	OpenOffice.org Spreadsheet template
`.dif`	Data Interchange Format
`.dbf`	dBASE/FoxPro database files
`.xls`	Microsoft Excel 97/2000/XP or Excel 95/5.0
`.xlt`	Microsoft Excel 97/2000/XP or Excel 95/5.0 template
`.sdc`	StarOffice Calc 5.0/4.0/3.0 (Sun StarOffice spreadsheet)
`.vor`	StarOffice Calc 5.0/4.0/3.0 template
`.slk`	Symbolic link format; includes formulas, and cell and file links
`.wks`	Lotus 1-2-3
`.csv`	Comma-separated format; a spreadsheet in a text file
`.html`	Web page

Comma-Separated Format

Spreadsheets and other data tables are often represented in a text file in comma-separated format. In other words, each of the values in the following line can be imported into consecutive cells in a row in a spreadsheet:

```
height, 60, 61, 44, 78, 56, 66
```

OpenOffice.org Draw

You can use OpenOffice.org Draw to manage files in various graphics formats, from AutoCAD files to bitmaps. In other words, OpenOffice.org Draw is a design tool that can be used by everyone who works with graphics, from design engineers to graphics designers.

You can start OpenOffice.org Draw by selecting Main Menu ➢ Office ➢ OpenOffice.org Draw or by running the `oodraw` command from a GUI terminal window. Figure 30.13 shows the GNOME desktop, with up2date in work.

Figure 30.13
OpenOffice.org Draw artwork

As you can see in the figure, OpenOffice.org Draw includes a wide variety of toolbars, some that allow you to manage color, as well as others that let you draw and add objects. The toolbars are described in Table 30.14.

Table 30.14: OpenOffice.org Draw Toolbars

Toolbar	Description
Color	Allows selection from a variety of colors
Function	Configures basic functions such as open, print, and undo
Hyperlink	Sets up access to web pages
Main	Allows zoom; insertion of objects, such as text and geometric shapes; alignment of objects; and so on
Object	Configures grid creation, text editing, rotation, color, and so on.
Option	Supports editing and drawing options for lines, including thickness and color

OpenOffice.org Draw works with many types of drawings, including the formats described in Table 30.15. As you can see, OpenOffice.org drawings can work from a number of different applications, including Microsoft Excel, StarOffice Calc, dBASE/FoxPro databases, and more. You can also set up OpenOffice.org Calc with text files in comma-separated format.

Table 30.15: OpenOffice.org Draw File Formats

Format	Description
`.sxd`	OpenOffice.org drawing
`.std`	OpenOffice.org drawing template
`.bmp`	Microsoft Windows bitmap
`.dxf`	AutoCAD Interchange Format
`.emf`	Enhanced metafile
`.eps`	Encapsulated PostScript
`.gif`	Graphics Interchange Format
`.jpg`	Joint Photographic Experts Group
`.met`	OS/2 metafile
`.pbm`	Portable bitmap
`.pcd`	Photo CD (Kodak)
`.pct`	Macintosh Pict drawing
`.pcx`	Zsoft paintbrush
`.pgm`	Portable gray map
`.png`	Portable Network Graphic

Continued on next page

Table 30.15: OpenOffice.org Draw File Formats

Format	Description
`.ppm`	Portable pixel map
`.psd`	Adobe Photoshop
`.ras`	Sun raster image
`.sda`	StarOffice 5.0 Draw
`.sdd`	StarOffice 3.0 Draw
`.sgf`	StarWriter graphics
`.sgv`	StarDraw 2.0 graphics
`.svm`	StarView metafile
`.tga`	Truevision Targa
`.tiff`	Tagged Image File Format
`.vor`	StarOffice 5.0/3.0 Draw template
`.wmf`	Microsoft Windows metafile
`.xbm`	X bitmap
`.xpm`	X pixmap

OpenOffice.org Impress

When you create a presentation, you're essentially creating a slide show. Presentation applications are basically specialized word processors with graphics, and they support a slide show to a large audience in a room or online. You can use OpenOffice.org Impress to build the same types of presentations as you might with other applications, such as Microsoft PowerPoint or StarOffice Impress.

You can start OpenOffice.org Impress by selecting Main ➢ Office ➢ OpenOffice.org Impress, or by running the `ooimpress` command from a GUI terminal window. Figure 30.14 illustrates a typical presentation start screen, ready for you to convince your colleagues to adapt Linux in the enterprise.

As you can see in the figure, OpenOffice.org Impress includes a wide variety of toolbars, some that allow you to manage color as well as others that allow you to draw, manage text, and add objects. Table 30.16 describes the toolbars.

Table 30.16: OpenOffice.org Impress Toolbars

Toolbar	Description
Color	Allows selection from a variety of colors
Function	Configures basic functions such as open, print, and undo
Hyperlink	Sets up access to web pages

Continued on next page

TABLE 30.16: OPENOFFICE.ORG IMPRESS TOOLBARS *(continued)*

TOOLBAR	**DESCRIPTION**
Main	Allows zoom; insertion of objects, such as text and geometric shapes; alignment of objects; and so on
Object	Configures grid creation, text editing, rotation, color, and so on.
Option	Supports editing and drawing options for lines, including thickness and color
Presentation	Lets you manage the design of each slide

FIGURE 30.14
OpenOffice.org Impress ready for a presentation

When you first start OpenOffice.org Impress, you'll see an AutoPilot Presentation wizard, which lets you start from a blank sheet, a presentation template, or an existing work. If you're creating a new presentation, OpenOffice.org Impress configures a slide design, output media, and basic presentation notes.

OpenOffice.org Impress works with other types of presentation formats, including those described in Table 30.17. As you can see, OpenOffice.org presentations can work with data from other applications, including Microsoft PowerPoint, StarDraw, StarImpress, and any application that can save in `.cgm` format.

TABLE 30.17: OPENOFFICE.ORG IMPRESS FILE FORMATS

FORMAT	**DESCRIPTION**
`.sxi`	OpenOffice.org Presentation
`.sti`	OpenOffice.org Presentation template

Continued on next page

Table 30.17: OpenOffice.org Impress File Formats *(continued)*

Format	Description
`.sxd`	OpenOffice.org drawing
`.ppt`	Microsoft PowerPoint 97/2000/XP
`.pot`	Microsoft PowerPoint 97/2000/XP template
`.sda`	StarDraw 5.0
`.sdd`	StarDraw 3.0/StarImpress 4.0/5.0
`.vor`	StarImpress 4.0/5.0 template

OpenOffice.org Writer

One of the banes of computing is dealing with the various word processing formats. You need converters to translate Microsoft Word documents to Corel WordPerfect documents and evern StarOffice Write documents. While converters are built into most word processing programs, including OpenOffice.org Writer, every word processing application includes special features that aren't always translated properly, if at all.

OpenOffice.org Writer does an excellent job. However, there are special features used by people in a number of industries—including publishing—that OpenOffice.org Writer does not handle properly. Nevertheless, OpenOffice.org Writer is good enough for most applications, businesses, and more.

You can start OpenOffice.org Writer by selecting Main Menu ➢ Office ➢ OpenOffice.org Writer, or by running the `oowriter` command from a GUI terminal window. New documents created in Open-Office Writer can include all of the features that you might find in Microsoft Word. Figure 30.15 illustrates a typical document.

As you can see in the figure, OpenOffice.org Writer includes three basic toolbars, as described in Table 30.18.

Table 30.18: OpenOffice.org Writer Toolbars

Toolbar	Description
Function	Configures basic functions such as open, print, and undo
Hyperlink	Sets up access to web pages
Main	Allows spell checking, zoom, insertion of objects such as text and geometric shapes, form creation, and so on
Object	Configures fonts, styles, formatting, highlighting, and color

OpenOffice.org Writer works with files from other word processors, including those described in Table 30.19.

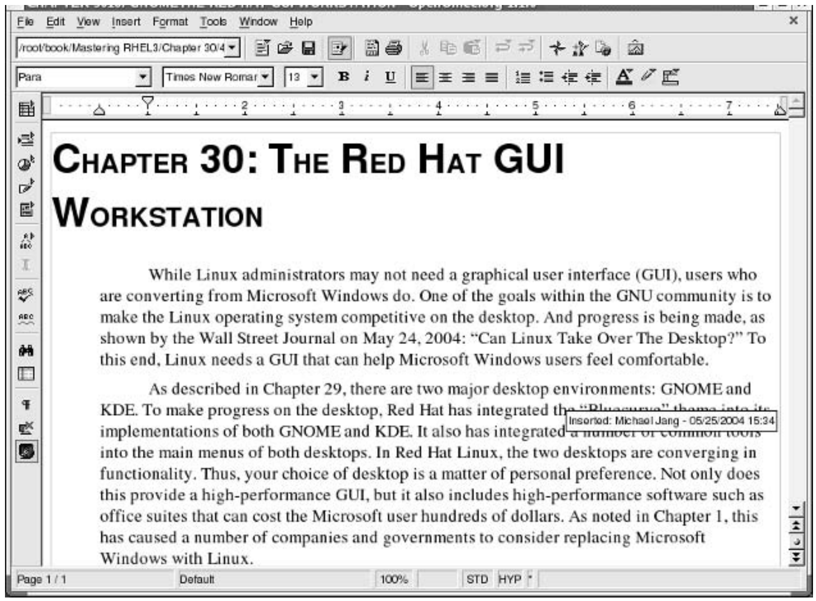


FIGURE 30.15
A Microsoft Word document in OpenOffice.org Writer

TABLE 30.19: OPENOFFICE.ORG WRITER FILE FORMATS

FORMAT	DESCRIPTION
`.sxw`	OpenOffice.org "text" document; it's not really text format.
`.stw`	OpenOffice.org "text" document template.
`.doc`	Microsoft Word 97/2000/XP; an alternate `.doc` format for Microsoft Word 95 and 6.0 is also available.
`.html`	Hypertext markup language, suitable for a web page.
`.rtf`	Rich Text Format; a relatively universal format readable by several word processors.
`.sdw`	StarWriter 3.0/4.0/5.0.
`.vor`	StarWriter 3.0/4.0/5.0 template.
`.txt`	Regular text; an alternate `.txt` format with coding for line breaks is also available.

NOTE *When more experienced Linux users need desktop publishing, they use text-based tools. For example, tools such as TeX and LaTeX include text commands that format titles, italics, and more in a text file. This is not unprecedented; even WordPerfect set up similar text commands through version 5.2.*

Other OpenOffice.org Tools

Other OpenOffice.org tools of note are:

- OpenOffice.org Math allows users to create and document equations of varying complexity; it supports trigonometric functions, integrals, limits, exponents, and more. You can start it from the command line with the `oomath` command.
- OpenOffice.org Printer Setup allows you to configure a driver and print format for the other parts of the OpenOffice.org suite. You can start it from the command line with the `oopadmin` command.

Opening Graphical Applications

Linux is well suited for graphics. Several major motion picture studios produce animations and special effects on Linux computers. With that in mind, it's worth exploring some of the graphical applications available for Linux.

A number of graphical applications come with Red Hat Enterprise Linux 3. They include PDF (Portable Document Format) readers, image viewers, and screen-capture programs. You can select most of these tools from the Main Menu ➢ Graphics and Main Menu ➢ Graphics ➢ More Graphics Applications submenus.

CrossOver Office

If you want to move to Linux but absolutely need those Microsoft applications, one option is CodeWeavers' CrossOver Office. For $39.95 (retail), it uses some of the work of the WINE (WINE is Not an emulator) project to let you run some of the most popular Microsoft Windows applications on your Linux computer. These applications include (but are not limited to):

- Microsoft Word 97/2000/XP
- Microsoft Excel 97/2000/XP
- Microsoft Outlook 97/2000/XP
- Microsoft PowerPoint 97/2000/XP
- Microsoft Visio
- Microsoft Internet Explorer
- Intuit Quicken
- Lotus Notes 5.0 and 6.51
- Adobe Photoshop

According to CodeWeavers, not all applications are perfectly compatible. Search its website to learn the current status of your desired applications. If you want an application to run as you may expect in Microsoft Windows, make sure the compatibility is at the Gold Medal level. Other applications may have significant bugs when you use CrossOver Office to run them under Linux. There are professional and standard versions of the CodeWeavers software are available. For additional information, navigate to `www.codeweavers.com`.

Graphical Document Readers

Three graphical document formats that you can read with Linux applications are PDF (Portable Document Format), PS (PostScript), and DVI (Device Independent).

While you can download Adobe Acrobat to read your PDF documents, Red Hat Enterprise Linux includes two native PDF readers: PDF Viewer and PS/PDF viewer. You can use the DVI Viewer, KViewShell, to read DVI documents.

PDF Viewer

To start the PDF Viewer, select Main Menu ➢ Graphics ➢ PDF Viewer, or run the `xpdf` command. This opens a simple screen with no toolbar; you can click the right mouse button to access some basic keyboard commands. If you want to open a PDF document, type **o**; in the Open dialog box shown in Figure 30.16, you'll be able to access open PDF files.

FIGURE 30.16
Accessing a PDF file

Once the file is open, you can use the arrow keys at the bottom of the screen to navigate through the document. Alternatively, you could use the basic commands listed in Table 30.20. Other commands are available; click the Question button for details.

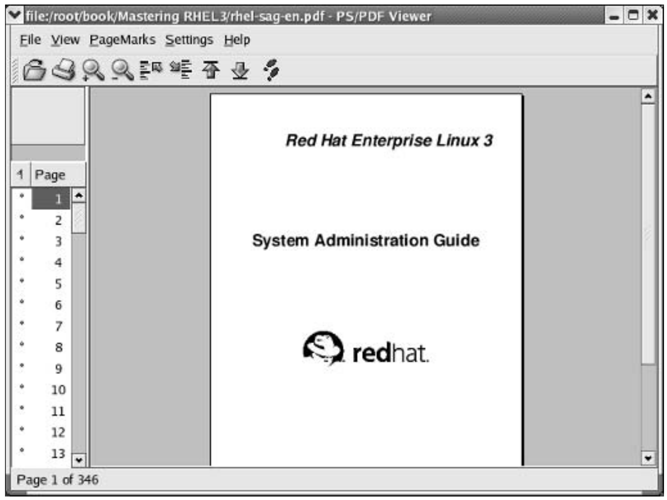
TABLE 30.20: BASIC XPDF COMMANDS

Command	Description
o	Opens a new file
f	Finds text
n	Goes to the following page
p	Moves to the previous page

The PS/PDF Viewer

The PS/PDF Viewer is also known as *KGhostView*. To start this application, select Main Menu ➤ Graphics ➤ More Graphics Applications ➤ PS/PDF Viewer, or run the `pdfviewer` command. This application provides a more intuitive interface for viewing both PDF and PS files, as shown in Figure 30.17.

Figure 30.17
Using KGhostView on a PDF file



The DVI Viewer

A number of Unix and Linux documents are processed from TeX to Device Independent (DVI) files. This DVI viewer is known as *KViewShell*. On the surface, it is similar to PDF, because it illustrates the GUI view of a typeset document.

To start this application, run the `kviewshell` *`filename`* command. For full functionality, this application requires the `tetex-*` RPM packages.

There are alternative DVI viewers. You can start KDVI by selecting Main Menu ➤ Graphics ➤ DVI viewer. You can start XDVI by selecting Main Menu ➤ Graphics ➤ More Graphics Applications ➤ DVI viewer

Note *TeX and LaTeX are formatting languages common in Linux and Unix, and are used to set up text files in a format suitable for publication.*

Image Viewers

The Red Hat Enterprise Linux GUI includes several image viewers. With some of these viewers, you can open, manipulate, and edit existing images. Each have different capabilities; Kuickshow sets up an image browser; the Icon Editor helps you manage the look and feel of icons within various GUI applications.

The Eye of GNOME

The Eye of GNOME is a graphics file viewer. It allows you to view images from a variety of file formats. Writing from this program is somewhat limited; by default you can write files only in JPEG and PNG formats.

To start this application, select Main Menu ➢ Graphics ➢ More Graphics Applications ➢ Eye of Gnome Image Viewer, or run the `eog` command.

The Icon Editor

The Icon Editor, KIconEdit, enables you to open and modify the look and feel of different icons. To start this application, select Main Menu ➢ Graphics ➢ More Graphics Applications ➢ Icon Editor, or run the `kiconedit` command.

The Image Viewer

The Image Viewer, KView, is another graphics file viewer similar to the Eye of GNOME. It supports output in many different image formats. To start this application, select Main Menu ➢ Graphics ➢ More Graphics Applications ➢ Image Viewer, or run the `kview` command.

Kuickshow

Kuickshow is an image browser that lists available images in the directory of your choice. To start this application, select Main Menu ➢ Graphics ➢ More Graphics Applications ➢ Kuickshow, or run the `kuickshow` command.

When you double-click on an image file, Kuickshow opens the image in its own window. You can then right-click on the image to open a menu that lets you manipulate the look and feel of the image.

Paint Program

The Paint Program, also known as KPaint, allows you to open, add to, and modify the images of your choice. To start this application, select Main Menu ➢ Graphics ➢ More Graphics Applications ➢ Paint Program, or run the `kpaint` command.

Screen-Capture Programs

Sometimes you'll want to record the settings on your screen. If you're describing a problem to someone, you can set up a picture that includes the look and feel of your desktop.

Some programs take their images from other hardware, such as digital cameras and scanners. Others take their images directly from a desktop screen or an active desktop window.

Digital Cameras

There are several GUI digital camera front ends to the `gphoto2-*` RPM. You can start the associated Linux digital camera application; select Main Menu ➢ Graphics ➢ Digital Camera Tool, or run the `gtcam` command. The list of cameras that it can detect is not complete; more information is available from `www.gphoto.org`.

Note *There are a couple of specialty HOWTO documents at* `www.tldp.org` *that may help: the Kodak-Digitalcam-HOWTO and the USB-Digital-Camera-HOWTO.*

Scanning

The standard Red Hat GUI scanning program is known as *xsane*, which you can start by selecting Main Menu ➢ Graphics ➢ Scanning, or by running the `xsane` command. Not all scanners are detected by xsane; in that case, you're prompted with this information, and xsane does not open.

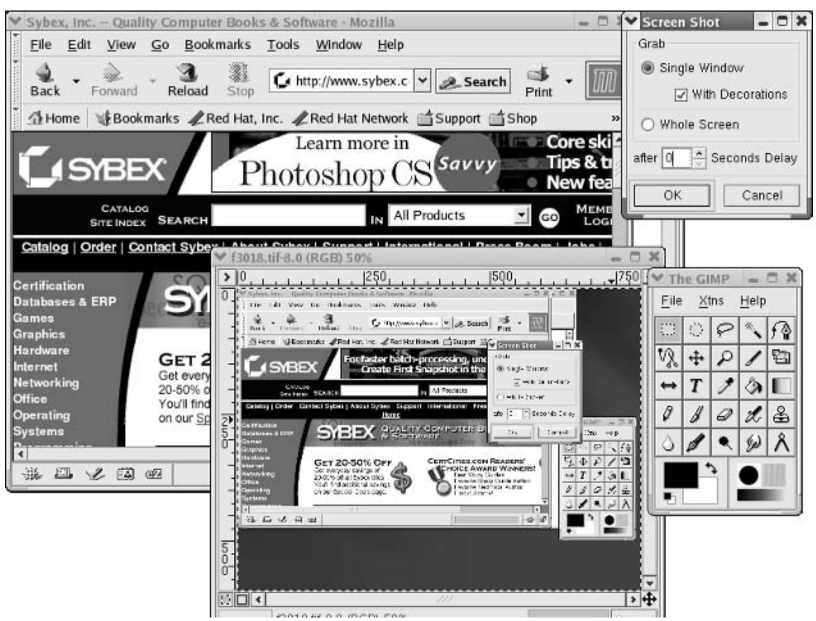
There is also a KDE scanning program, Kooka. It supports xsane and provides character-recognition functions. You start Kooka by selecting Main Menu ➢ Graphics ➢ More Graphics Applications ➢ Scan & OCR Program or by issuing the `kooka` command.

The GIMP

My favorite Linux graphics program is The GIMP, which is the GNU Image Manipulation Program. Many Linux users prefer The GIMP to other high-end image programs, such as Adobe's Photoshop and Jasc's Paint Shop Pro. It's a part of the GNOME office suite. I've used it to configure most of the artwork for this book. To start this application, select Main Menu ➢ Graphics ➢ The GIMP, or run the `gimp` command.

For example, when I took a screenshot of the Mozilla browser, I started The GIMP, then selected File ➢ Acquire ➢ Screen Shot. This opened the Screen Shot window. Once Mozilla was ready, I clicked OK in the screenshot window; this turned the cursor into a plus sign (+). I used it to select the KPPP window. When I right-clicked the screenshot, it opened a menu that I used to save the image in the file of my choice. The various screens and result are shown in Figure 30.18.

Figure 30.18
The GIMP at work



Another Graphical Program: Color Chooser

KColorChooser, also known as KColorEdit, allows you to edit color palettes. To start this application, select Main Menu ➤ Graphics ➤ More Graphics Applications ➤ Color Chooser, or run the `kcolorchooser` command. In this utility, you can measure the relative levels of RGB (red, green, and blue); every color has different levels of red, green, and blue between 0 and 255.

Setting Default Languages

You can configure the GUI in one of a number of languages. This process is easy if you've installed the desired language during the installation process. All you need to do is select the desired language using the Language Selection tool. But if you're working as an international enterprise, you may need one more language.

Basic Configuration Files

If you need one more language, you can install the required configuration files. For a list, refer back to the `comps.xml` configuration file described in Chapter 5. As an example, let's assume we want to add Korean language configuration files to a computer. Take a look at the Korean Support section of the `comps.xml` file:

```
<group>
  <id>korean-support</id>
  <uservisible>false</uservisible>
  <name>Korean Support</name>

  <langonly>ko_KR</langonly>
  <packagelist>
    <packagereq type="optional" requires="kdelibs">kde-i18n-Korean</packagereq>
    <packagereq type="optional" requires="man-pages">man-pages-ko</packagereq>
    <packagereq type="optional" requires="XFree86">ami</packagereq>
    <packagereq type="mandatory">h2ps</packagereq>
    <packagereq type="mandatory">nhpf</packagereq>
    <packagereq type="mandatory">ttfonts-ko</packagereq>
  </packagelist>
</group>
```

You can see the RPM packages associated with this package group. If you want to add Korean language support, you'll want to install these packages on your system.

Next, open `/etc/sysconfig/i18n`. Modify the `SUPPORTED` variable for the appropriate locale. Different languages and character sets are listed in the `/usr/X11R6/lib/X11/locale` directory. For example, the associated language locale and character type for Korean in that directory is

```
ko_KR.UTF-8
```

If you don't find your language in this directory, look in the `locale.*` files in that directory. Now open the `/etc/sysconfig/i18n` file. Add the desired locale to the `SUPPORTED` variable in the following format:

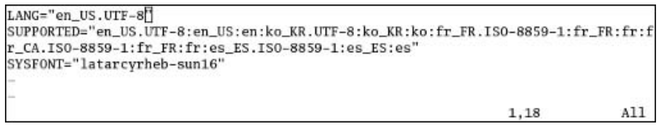
```
language_locale.chartype:language_locale:language
```

For the listed Korean language, locale, and character type, that is

```
ko_KR.UTF-8:ko_KR:ko
```

You can see the result in Figure 30.19, which shows my `/etc/sysconfig/i18n` file. This file includes settings for French (variants for Canada and France), U.S. English, Korean, and Spanish.

FIGURE 30.19
/etc/sysconfig/i18n language settings



Red Hat Language Selection Tool

You can use the Red Hat language utility to select a graphical default from the languages that you have installed. Start this utility by selecting Main Menu ➢ System Settings ➢ Language, or run the `redhat-config-language` command from a GUI command-line interface. This opens the Language Selection window, shown in Figure 30.20. The window normally includes the languages that you included during the installation process.

FIGURE 30.20
Setting the default language

The Korean language option should now appear the next time you open the `redhat-config-language` utility window. When you select a different language, `redhat-config-language` tells you that the changes will take effect the next time you log in. Figure 30.21 illustrates the result, a Korean language version of the KDE desktop. Notice how Evolution is also illustrated in Korean.

Some applications require their own language packages and settings, which is beyond the scope of `redhat-config-language` and this book. Several language-specific HOWTOs are available through the Linux Documentation Project (`www.tldp.org`) that may be able to help.

Just to show how easy it is to switch languages, I've repeated this process with the Spanish language. You can see the result in the GNOME desktop in Figure 30.22.

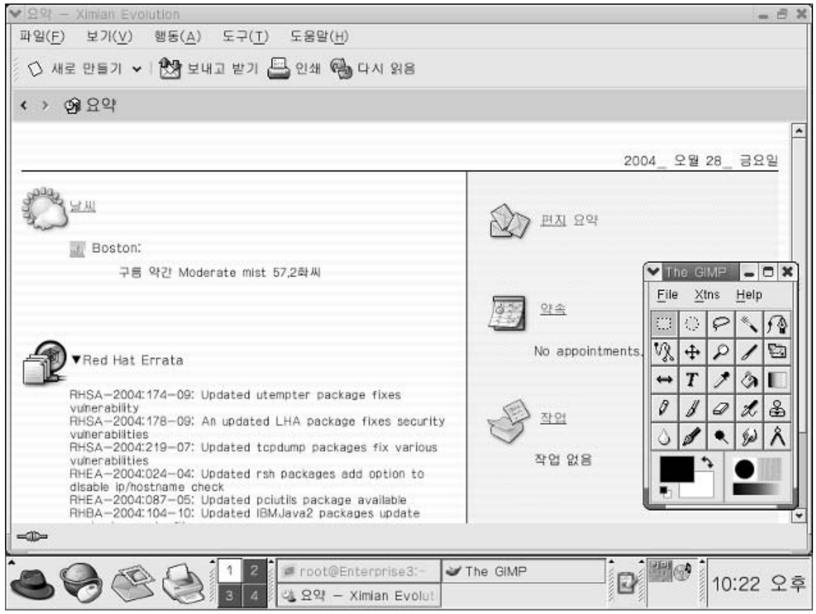


FIGURE 30.21 The KDE desktop in Korean

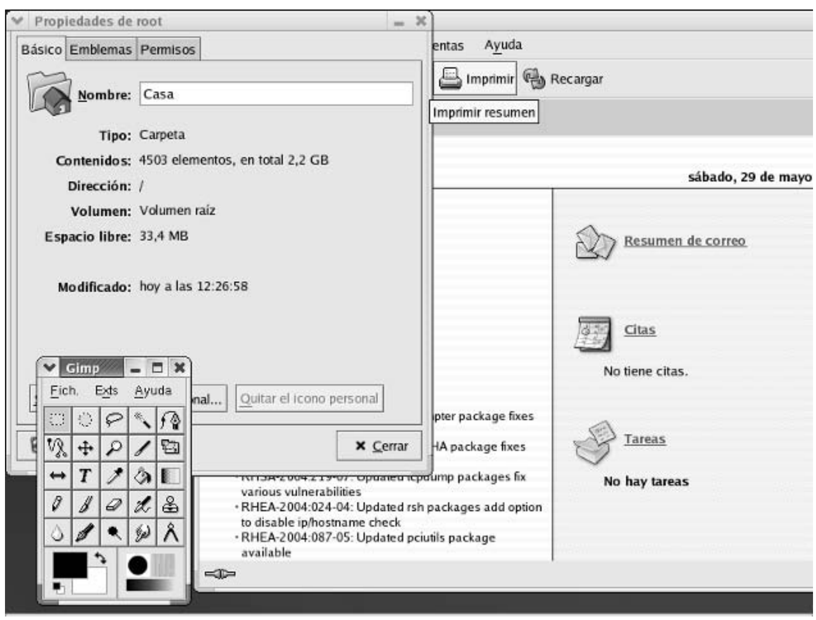


FIGURE 30.22 The GNOME desktop in Spanish

Summary

This has been a basic introduction to the Linux GUI desktop. Red Hat Enterprise Linux allows you to install both the GNOME and KDE desktops. Both of these desktops include many of the same tools that you might find in Microsoft Windows, and more. In fact, with Red Hat's Bluecurve theme, both desktops have a similar look and feel on Red Hat Enterprise Linux. It's easy to configure either the GNOME or KDE desktop as a workstation to meet the needs of your users.

A substantial number of extras are available through GNOMEand KDE. This includes a wide variety of software that could easily cost you hundreds of dollars. It includes the accessories that you need every day. The Internet applications consist of browsers, e-mail managers, and chat clients. The Sound and Video utilities allow you to manage, process, and record multimedia. Both desktop environments includes a number of system tools that help administrators manage their systems.

The office suite included with Red Hat Enteprise Linux is OpenOffice.org. This suite includes a word processor, spreadsheet, drawing program, diagram creator, and presentation manager.

The graphical programs run the gamut from simple color managers to fully featured graphical programs. GUI documentation viewers, such as PDF, DVI, and PS readers, fall into the same category.

Finally, you can add a few packages as defined in `comps.xml` to adapt your GUI to the languages of your choice. Once configured, you can use the Language Selection tool to set a new default language for your GUI desktop environment.

Appendices

In this section, you will learn how to:

- **Appendix A: More Information Online**
- **Appendix B: The GNU General Public License**

Appendix A

More Information Online

IF YOU'VE PURCHASED THE subscription to Red Hat Enterprise Linux 3, your first resource is Red Hat Support. Contact information is available with your subscription. Sales information is available at `www.redhat.com` or 1-888-REDHAT-1.

A wealth of information about Linux is available online. That's not a surprise, since Linux is developed through the cooperation of people working together from around the world. They shared what they learned online, and all of us can benefit from their experience.

If the documentation is not enough, several excellent newsgroup "libraries" are available that can help you find many of the answers you need. If you still can't find the answer, and you can show that you've done your "homework," you'll find many people in these newsgroups who are ready to help solve difficult problems. There are also Linux user groups available worldwide where you can share and learn more about Linux in person.

When you find solutions, you may need to download new packages. You can download the latest utilities, software, and kernels from several web and FTP sites.

As Linux develops, there is a constant stream of news on this operating system. Various Linux certifications are available if you want to prove your credentials to the rest of the world. An almost endless number of applications are constantly being improved for Linux. And when you have hardware questions, you can find websites dedicated to making Linux work with various components in the PC.

Neither Sybex nor I endorse or sponsor any of these web, newsgroup, or mailing list sites. The lists in this appendix are far from comprehensive. We include them simply as an aid in your research. The axiom *caveat emptor,* let the buyer beware, applies to all these sites as well.

NOTE *Web links may change by the time you read this book. If the link does not work, you'll have to use your own insight on the Internet to find the information associated with the noted website.*

This appendix is just a brief list of Linux links; many more are available online from sites such as `www.linuxlinks.com`. This appendix is organized by and therefore covers the following types of Internet sites:

- Online Linux documentation
- Linux newsgroups and mailing lists

- Download sites
- Linux news
- Professional certifications
- Linux applications
- Linux hardware
- General information

Online Linux Documentation

Perhaps the first word in Linux documentation is the Linux Documentation Project, available online at `www.tldp.org`. It includes the HOWTOs on many Linux topics, book-length guides, FAQs, and man pages. A number of other websites include copies, or mirrors, of the HOWTOs. While this is far from a comprehensive list, there are several other great sources of information, briefly described in Table A.1.

Table A.1 is just a sample of the available websites dedicated to Linux and related software. My selection is based solely on the sites I know and use, and any significant omissions are not intentional.

TABLE A.1: ONLINE LINUX DOCUMENTATION

SITE	URL	DESCRIPTION
ApacheWeek	`www.apacheweek.com`	An online journal for the Apache web server software.
Free Software Foundation	`www.fsf.org`	The developers behind a lot of the original Linux software and the GNU Public License.
Just Linux	`www.justlinux.com`	An online forum for Linux news and discussion.
Linux Documentation Project	`www.tldp.org`	The repository for a wide variety of Linux documentation, including the HOWTOs.
Linux Focus	`www.linuxfocus.org`	A multilingual quarterly international Linux magazine.
Linux Forum	`www.linuxforum.com`	An online news and message forum site, courtesy of the WebFreaks.
Linux Gazette	`www.linuxgazette.com`	An online publication dedicated to "sharing ideas and discoveries."
Linux Hardware	`www.linuxhardware.org`	Provides in-depth coverage of hardware that works for Linux.
Linux Journal	`www.linuxjournal.com`	One of the first journals on Linux, owned by Specialized System Consultants.

Continued on next page

Table A.1: Online Linux Documentation *(continued)*

Site	URL	Description
Linux Kernel 2.4 Internals	`tldp.org/LDP/lki`	An introduction to the Linux 2.4 kernel, only available online.
Linux Magazine	`www.linux-mag.com`	Another Linux magazine, also available in bookstores.
Linux Magazine	`www.linux-magazine.com`	A European Linux magazine, printed in the UK, also available in many US bookstores.
Linux Network Administrator's Guide	`tldp.org/LDP/nag2`	An online version of the O'Reilly book of the same name.
Linux Planet	`www.linuxplanet.com`	An online news magazine dedicated to news, reviews, tutorials, and more.
Linux Questions	`linuxquestions.org`	Primarily a forum for Linux questions and answers.
Linux System Administrator's Guide	`tldp.org/LDP/sag`	An online version of the O'Reilly book of the same name.
Maximum RPM	`www.redhat.com/docs/books/max-rpm/`	An older version of the still-valuable guide to the Red Hat Package Manager.
Red Hat Documentation	`www.redhat.com/docs`	Includes online manuals for various Red Hat operating systems, as well as links to various books.
SearchEnterpriseLinux	`searchenterpriselinux.com`	An online journal from TechTarget.com.
Sys Admin	`www.samag.com`	The self-described "journal for Unix systems administrators" also has good tips for Linux users.
Wide Open Magazine	`redhatmagazine.com`	Subscription interface to Red Hat's Linux magazine. Free for qualifying users.
Computer Power User Magazine	`www.computerpoweruser.com`	Includes a number of Linux related hardware articles; click Search All Articles to find what you need.

Linux Newsgroups and Mailing Lists

Linux is under constant development by a community. Many members of that community are anxious to make their name by solving new problems, and their insights are available online. It's quite possible that the answer to your problem is already available in the Internet newsgroup database, accessible through `groups.google.com`.

Alternatively, you can monitor individual newsgroups or subscribe to various mailing lists. Many mailing lists are available for specific distributions and applications, as well as through Linux user groups.

Tip *It's an excellent idea to subscribe to the Red Hat Enterprise Linux 3 mailing list at* `www.redhat.com/mailman/listinfo/taroon-list`*. You can get help here from people who are dedicated to the operating system associated with this group. A Red Hat Enterprise subscription is not required for membership, as of this writing. However, it is not intended as a substitute for official Red Hat support.*

If you choose to post on a Linux newsgroup, be careful. Many newsgroups are dedicated to specific topics, which may lead to answers unrelated to your posts. Others use the e-mail addresses that they find on newsgroups for advertising, a practice known as *spamming*.

A wide variety of newsgroups are available through your newsreader, as shown in Table A.2. Be picky; while some Linux newsgroups don't get a lot of valuable traffic, many are worth browsing on a regular basis.

Table A.2: Some Linux Newsgroups

Newsgroup	Description
`alt.linux`	An active group.
`alt.os.linux`	An active group, focused on the operating system.
`alt.os.linux.*`	Several groups are available for different distributions such as `alt.os.linux.redhat`.
`at.linux`	A Linux newsgroup in German (Austria).
`comp.os.linux`	Another active newsgroup.
`comp.os.linux.*`	Several different newsgroups, including those on different CPUs, hardware, networking, security, the X Window, and more.
`cz.comp.linux.*`	Linux newsgroups in Czech; the Red Hat newsgroup is `cz.comp.linux.redhat-cz`.
`de.comp.os.unix.linux.*`	Linux newsgroups in German.
`es.comp.os.linux.*`	Linux newsgroups in Spanish.
`esp.comp.so.linux.*`	More Linux newsgroups in Spanish.
`fido?.*.linux`	Several Linux newsgroups in different languages; for example, `fido7.ru.unix.linux` is a Russian-language newsgroup.
`fj.os.linux.*`	Linux newsgroups in Japanese.
`fr.comp.os.linux.*`	Linux newsgroups in French.
`han.comp.os.linux.*`	Linux newsgroups in Korean.
`hun.lists.mlf.linux*`	Linux newsgroups in Hungarian.

Continued on next page

Table A.2: Some Linux Newsgroups *(continued)*

Newsgroup	Description
it.comp.os.linux.*	Linux newsgroups in Italian.
linux.apps.*	A wide variety of Linux newsgroups on various types of applications; many are not active.
linux.debian.*	Many Linux newsgroups related to the Debian Linux distribution.
linux.dev.*	A wide variety of Linux newsgroups on various devices and drivers.
linux.redhat.*	Many Linux newsgroups related to Red Hat Linux.
nl.comp.os.linux.*	Linux newsgroups in Dutch.
no.it.os.unix.linux.*	Linux newsgroups in Norwegian.
pl.comp.*.*	Linux newsgroups in Polish.
vmware.*.*	Newsgroups related to the VMware virtual machine software; more are available through the VMware newsgroup server at `news.vmware.com`.

Several Linux mailing lists are available online as well. Red Hat has a wide variety of mailing lists that you can subscribe to at

`www.redhat.com/mailing-lists`

As described earlier, there is a standard mailing list for Red Hat Enterprise Linux 3. Other active Red Hat mailing lists are available for different applications and services, such as Apache, Samba, and CUPS. You can sign up through the links noted on the aforementioned Web page. Navigate to their web pages listed later in this appendix for more information.

Depending on the communities, you may find forum-based help associated with the "rebuilds." The Community Linux (cAos) folks have a mailing list as well as IRC channels for real-time discussion. White Box Enterprise Linux includes some message board–style forums at `whiteboxlinux.net/forum.php`. Tao Linux is a self-described "community supported" rebuild, with mailing lists available through `taolinux.org/?q=node/view/10`.

Linux user groups (LUG) commonly maintain their own mailing lists for their users. It can be helpful to join one in your local area. People are more likely to help you if they know your face. LUGs are available all over the world. You may be able to find a LUG in your area through one of the websites noted in Table A.3.

Table A.3: Linux User Group Lists

Site	Description
www.linux.org/groups	The Linux Online user groups site
www.ssc.com:8080/glue	Groups of Linux Users Everywhere
www.redhat.com/opensourcenow	Red Hat's user group and open-source advocacy program

Download Sites

The official installation packages for Red Hat Enterprise Linux are available only via paid subscription. However, the source code for each package is available from the Red Hat FTP site (and mirrors).

As described in Chapter 1, several third parties have compiled this source code into usable RPMs. To comply with Red Hat trademark limitations, they have replaced items such as the Main Menu icon (the red Fedora). In most cases, they are organized in ISO files, which you can download and then use to install.

You can also download a number of Linux distributions that you can install with the `rpm` or `tar` commands described in Chapter 10. Some of the most popular download sites are listed in Table A.4.

While it's often more convenient to download from an HTTP site, FTP downloads are usually faster. As explained in Chapter 22, this is because FTP is built for file transfers. Many of the websites listed in the table include FTP links for download.

Table A.4: Downloading Linux

Site	URL	Description
Red Hat	`ftp.redhat.com`	Requires anonymous access; often busy.
Community Linux	`www.caosity.org`	A community-based "rebuild" of Red Hat Enterprise Linux 3; my personal favorite.
Freshmeat	`www.freshmeat.net`	Offers the latest in Linux software; includes development home pages and FTP download links for numerous Linux components.
ibiblio Linux Archive	`www.ibiblio.org/pub/Linux`	Features an archive with more than 170GB of Linux software; from the University of North Carolina.
LinuxApps	`products.enterpriseitplanet.com/linux.html`	A comprehensive download source for Linux applications.
Linux ISO	`www.linuxiso.org`	A site where you can download `.iso` files for Linux distributions; with `cdrecord`, you can turn them into Linux installation CDs. Download links use FTP servers. Unfortunately, links to the rebuilds are not currently available.
The Linux Kernel Archives	`ftp.kernel.org/pub`	Includes the latest stable, patch, and beta versions of the Linux kernel. Note: These may not include the features in Red Hat's custom Enterprise kernels.

Continued on next page

TABLE A.4: DOWNLOADING LINUX *(continued)*

SITE	URL	DESCRIPTION
RPM Find	`www.rpmfind.net`	A comprehensive database of available RPM packages for a variety of distributions.
Source Forge	`sourceforge.net`	The self-described "world's largest open-source software development website"; includes development home pages and FTP download links for numerous Linux components.
Tao Linux	`taolinux.org`	A "rebuild" developed through the Alfred University (NY) Linux Users' Group.
Tucows Linux	`linux.tucows.com`	An all-in-one site for Linux downloads; you can select a mirror close to you before starting the download.
White Box Linux	`whiteboxlinux.org`	A "rebuild" initially developed through the Beauregard Parish library in Louisiana.

The Red Hat FTP site can be especially busy; you may want to try your download from one of the Red Hat mirror sites available around the world. The official list is available at `www.redhat.com/download/mirror.html`.

You can find a huge list of sites with downloadable Linux software at `www.linuxbasis.com/downloads.html`.

Linux News

Linux is developing every day. If you need the latest Linux software, whether it is for new features, for security enhancements, or just to be "cool" in the Linux community, read some of the sites listed in Table A.5 on a regular basis.

TABLE A.5: LINUX NEWS SITES

SITE	URL	DESCRIPTION
Linux Insider	`www.linuxinsider.com`	A listing of the latest news stories on Linux online; similar to Linux Today.
Linux Online News	`www.linux.org/news`	A listing of the latest news stories on Linux online; similar to Linux Today.

Continued on next page

Table A.5: Linux News Sites *(continued)*

Site	URL	Description
Linux Planet	`www.linuxplanet.com`	A resource of in-depth articles on the latest Linux software.
Linux Today	`www.linuxtoday.com`	In my opinion, the premier site for Linux news and information; links to news stories from other sites.
Linux Weekly News	`lwn.net`	A weekly review of the latest Linux developments.
NewsForge	`newsforge.com`	A listing of the latest news on Linux; part of the Open Source Development Network.
Slashdot	`www.slashdot.org`	Self-described as "News for Nerds. Stuff that matters."

Professional Certifications

There are four major Linux certification programs, which we discussed in detail in Chapters 27 and 28. If you're considering one of these certifications, check them frequently; I've seen exam updates as frequently as 6–12 months. They are summarized in Table A.6.

Table A.6: Linux Certification Programs

Program	URL	Description
Linux+	`www.comptia.org`	Entry-level certification from CompTIA; intended for users with 6–12 months of Linux experience.
Linux Professional Institute	`www.lpi.org`	Midlevel certifications from a nonprofit organization; not affiliated with any distribution. Two levels of exams are available at this time.
Red Hat	`www.redhat.com`	Offers the Red Hat Certified Engineer (RHCE), Red Hat Certified Technician (RHCT) and Red Hat Certified Architect (RHCA) exams; all are hands-on; the RHCA is considered to be one of the most difficult and practical exams in the computer industry.
SAIR Linux and GNU	`www.sairinc.com`	Midlevel certifications; not affiliated with any distribution. Two levels of exams are available at this time. Affiliated with Thomson/Course Technology.

Linux Applications

You can find a wide variety of websites for just about every current Linux application. A few of them are listed in Table A.7. They vary widely in content; many include documentation and downloads of the latest versions of the software. Some applications include an open-source and a commercial version.

Table A.7: Linux Applications

Application	URL	Description
Amanda	`www.amanda.org`	The Advanced Maryland Automatic Network Disk Archiver; for backups.
Apache	`httpd.apache.org`	The most popular web server on the Internet.
Code Weavers	`www.codeweavers.com`	Their CrossOver office product runs several Microsoft Windows applications, including Microsoft Office 2000, Quicken, and Lotus Notes.
Common Unix Printing System	`www.cups.org`	The default print server for Red Hat Linux; a commercial version is available from Easy Software Products at `www.easysw.com`.
DNS/BIND	`www.isc.org/products/BIND`	The Domain Name System server software is based on the Berkeley Internet Name Domain (BIND).
The GIMP	`www.gimp.org`	The GNU Image Manipulation Program is a fully featured image manager, similar to Paint Shop Pro.
GNOME	`www.gnome.org`	The GNU Network Object Model Environment is from a group that develops a wide variety of applications.
Houdini	`www.sidefx.com`	The proprietary graphics software used by some movie studios.
KDE	`www.kde.org`	The K Desktop Environment is from a group that develops a wide variety of applications.
LDAP	`www.openldap.org`	Home of the open-source Lightweight Directory Access Protocol server.
Linspire (Lindows)	`www.linspire.com`	An operating system that incorporates proprietary technologies to run Microsoft Windows software inside a Linux X Window.
MySQL	`www.mysql.com`	The open-source database program commonly associated with Linux.
OpenOffice.org	`www.openoffice.org`	A group dedicated to creating an open-source office suite; the default for Red Hat Enterprise Linux. Versions are also available for Microsoft Windows.

Continued on next page

Table A.7: Linux Applications *(continued)*

Application	URL	Description
OpenSSH	www.networksimplicity.com	The developers of the Secure Shell software; a version is available for Microsoft Windows.
Samba	www.samba.org	The software that allows Linux and Unix-style computers to work on a Microsoft Windows–based network.
Sendmail	www.sendmail.com	The commercial version of the mail server described in Chapter 20. The open-source version is at www.sendmail.org.
Star Office	wwws.sun.com/software/star/staroffice/6.0/	An office suite developed by Sun Microsystems that works with Linux, other Unix-style operating systems, and Microsoft Windows.
Transgaming	www.transgaming.com	The developers of cross-platform gaming technologies.
Tripwire	www.tripwire.com	The developers of software for checking the security of a network.
VMware	www.vmware.com	The developers of the virtual machine application that allows you to run Linux on Microsoft Windows (and vice versa).
Win4Lin	www.trelos.com	The developers of the virtual machine application that allows you to run Microsoft Windows on Linux.
XFree86	www.xfree86.org	The XFree86 Project, developers of the standard Linux X Window software.
X.Org Foundation	www.x.org	The X Project; developers of the X Window software used on the current version of Fedora Linux.

Linux Hardware

Several groups are dedicated to making it easy to work with every type of hardware on Linux. Some of these hardware groups are described briefly in Table A.8.

TABLE A.8: LINUX HARDWARE GROUPS

HARDWARE	URL	DESCRIPTION
Digital Cameras	`www.gphoto.org`	Provides software for various digital camera interfaces.
FireWire (IEEE1394)	`www.linux1394.org`	Supports IEEE 1394 hardware; interfaces are still "experimental" in the Red Hat Linux kernel.
Laptop computers	`www.linux-laptop.net`	The Linux on Laptops site provides tips for a wide variety of makes and models of laptop, notebook, and palmtop computers.
The Linux-Mobile Guide	`tuxmobil.org/howtos.html`	Provides tips for configuring mobile computers, including laptops and palmtops.
Linux Network Drivers	`www.scyld.com/community.html`	Includes the latest Ethernet network drivers.
The Linux Printing Database	`www.linuxprinting.org`	A resource for print drivers.
Modems	`www.linmodems.org`	The work of the Linux Winmodem Support group is helping Linux work with many of these proprietary modems.
Scanner Access Now Easy	`www.sane-project.org`	A resource for using scanners on Linux.
Sound Cards	`www.alsa-project.org`	The Advanced Linux Sound Architecture (ALSA) project provides audio and MIDI support.
USB	`www.linux-usb.org`	The Linux USB Project is constantly releasing new drivers in support of new USB devices.

General Information

You can find general information about Linux at a number of basic websites. In Table A.9, we've included the websites of several of the other major Linux distributions, because they are repositories of good information..

TABLE A.9: GENERAL LINUX INFORMATION

SITE	URL	DESCRIPTION
Bastille Linux	`www.bastille-linux.org`	A system designed to secure a number of different Linux and Unix distributions. Updated to support Red Hat Enterprise Linux 3.
Conectiva Linux	`www.conectiva.com.br`	A Linux distribution based in Brazil; originally developed from Red Hat Linux.
Debian Linux	`www.debian.org`	A Linux distribution developed entirely by volunteers.
Just Linux	`www.justlinux.com`	A great resource for newer Linux users, with guides and articles on basic Linux operations, formerly known as Linux Newbie.
Linux.com	`www.linux.com`	A Linux portal with links to NewsForge, Documents, and Freshmeat software.
Linux Online	`www.linux.org`	A Linux portal with documents, news, downloads, reviews, and more.
Security Enhanced Linux	`www.nsa.gov/selinux`	A revised kernel developed by the U.S. National Security Agency; many of its features are part of Red Hat Enterprise Linux.
SUSE Linux	`www.suse.com`	A Linux distribution with a big following in Europe; now part of Novell.
Turbolinux	`www.turbolinux.com`	A Linux distribution with a big following in Asia; now part of the United Linux consortium.
Xandros	`www.xandros.com`	The developers of a desktop version of Linux that has a Microsoft Windows "look and feel"; developed from the former Corel Linux distribution.

Appendix B

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

2. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   A. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

11. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

1. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

2. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Index

Note to the Reader: Throughout this index **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

C

D

F

G

H

L

N

O

Y

Z